

10

جاوا اسکرپٹ کی مدد سے کلائنٹ-سائڈ اسکرپٹنگ

مواد کی تخلیق پیچیدہ نہیں ہونی چاہیے۔ یہ اس قدر انوکھی/غیر معمولی نہیں ہونی چاہیے کہ جسے صرف کمپیوٹر سائنس کے ماہرین ہی انجام دے سکیں۔
برینڈن ایچ
جاوا اسکرپٹ کے تخلیق کار

تعارف

باب 9 میں ہم نے سیکھا ہے کہ HTML کا استعمال کر کے ویب صفحہ کی تشکیل کس طرح کی جاتی ہے۔ HTML دستاویز کئی ٹیکس (Tags) پر مشتمل ہوتا ہے جو HTML کو یہ بتاتے ہیں کہ متن اور گرافکس کو کس طرح ظاہر کیا جائے گا۔ ہم نے سکونی (Static) ویب صفحات (جو مکالماتی نہیں ہیں) کی تشکیل کا طریقہ بھی سیکھا ہے۔ HTML دستاویز کو مکالماتی بنانے کے لیے ضروری ہے کہ HTML میں کچھ مخصوص کوڈ (Scripts) شامل کیے جائیں۔ جاوا اسکرپٹ (Java Script) اسی قسم کی اسکرپٹنگ لینگویج ہے۔

اس باب میں ہم جاوا اسکرپٹ سے متعلق بنیادی باتوں کو سیکھیں گے اور ساتھ ہی یہ بھی سیکھیں گے کہ ایکسپریشن (عبارتوں)، آپریٹرز پوپ اپ باکس (Pop up Box)، کنورژن فنکشن، مشروط بیانات (Conditional Statement)، لوپنگ اسٹیٹمنٹ، آہجیکٹ میں رد و بدل سے متعلق بیانات اور جاوا اسکرپٹ فنکشن کا استعمال کر کے ویب صفحات میں متحرک اثرات کس طرح شامل کیے جائیں۔

10.1 جاوا اسکرپٹ کے بارے میں (About JavaScript)

جاوا اسکرپٹ کی تخلیق برینڈن ایچ (Brendan Eich) نے کی اور یہ ستمبر 1995 میں

مقاصد

- اس باب کو مکمل کرنے کے بعد طلباء اس لائق ہو جائیں گے کہ:
- جاوا اسکرپٹ کی تعریف بیان کر سکیں،
- جاوا اسکرپٹ سے متعلق بنیادی باتوں کی وضاحت کر سکیں،
- HTML دستاویز میں جاوا اسکرپٹ کوڈ داخل کر سکیں،
- Declare اور متغیرات (Variable) کے استعمال کے درمیان موازنہ کر سکیں،
- عبارتوں (Expression) میں متغیرات اور حرفی علامتوں کا استعمال کر سکیں،
- مختلف ڈیٹا ٹائپ (Data Types) اور قدروں (Values) کو بیان کر سکیں،
- برانچنگ (Branching) اور لوپ اسٹیٹمنٹ کے استعمال کی اہمیت کو سمجھ سکیں،
- for لوپ کے ساتھ تکرار کا عمل انجام دے سکیں،
- while اور do..while لوپ کے درمیان فرق کر سکیں،
- لوپ میں توقف (Break) اور تسلسل پیدا کر سکیں،
- آہجیکٹ میں رد و بدل سے متعلق کچھ بیانات (Statement) سے بحث کر سکیں اور
- فنکشن کو متعین (Define) کرنے اور طلب کرنے کے طریقہ پر غور کر سکیں۔

جاوا اسکرپٹ کی مدد سے کلائنٹ سائڈ اسکرپٹنگ

اس وقت وجود میں آچکی تھی جب Netscape 2.0 (ایک ویب براؤزر) کو جاری کیا گیا تھا۔ جاوا اسکرپٹ ویب صفحات کو متحرک اور زیادہ مکالماتی بنانے کے مقصد سے ڈیزائن کیا گیا تھا۔ جاوا اسکرپٹ ایک مشہور و معروف اسکرپٹنگ لینگویج ہے۔ اس میں مندرجہ ذیل خصوصیات پائی جاتی ہیں۔

- اسے کلائنٹ اور سرور اپیلی کیشن کے لیے استعمال کیا جاسکتا ہے۔
- یہ پلیٹ فارم سے متشی لینگویج ہے اس کا مطلب یہ ہے کہ اسے کسی بھی آپریٹنگ سسٹم (یعنی Linux، مائکروسافٹ ونڈوز، Mac OS X وغیرہ) پر چلایا جاسکتا ہے۔
- جاوا اسکرپٹ کوڈ کو HTML دستاویز میں داخل کیا جاتا ہے یا دستاویز میں اس کا حوالہ دیا جاتا ہے۔ صرف ایسا کرنے کے بعد ہی اسے ویب براؤزر پر چلایا جاسکتا ہے۔
- یہ ایک تشریح شدہ لینگویج ہے۔
- یہ ایک case-sensitive لینگویج ہے اور اس کے کلیدی الفاظ صرف چھوٹے حروف میں لکھے جاتے ہیں۔

10.1.1 جاوا اور جاوا اسکرپٹ کے درمیان فرق

کچھ لوگ یہ سوچتے ہیں کہ جاوا اور جاوا اسکرپٹ ایک ہی ہیں جبکہ دونوں لینگویج ایک دوسرے سے بالکل مختلف ہیں۔ جاوا، سن مائکرو سسٹم کی عمومی مقصد کی معروض مرکزی (Object Oriented) پروگرامنگ لینگویج ہے جبکہ جاوا اسکرپٹ آبجیکٹ پر مبنی اسکرپٹنگ لینگویج ہے۔ اسکرپٹ مختصر پروگرامنگ اسٹیمینٹ ہے جو کسی مخصوص کام کو انجام دیتی ہے۔

10.1.2 جاوا اسکرپٹ کے ورژن

جاوا اسکرپٹ اور ویب براؤزر کے کچھ ورژن ذیل میں دیے گئے ہیں:

جاوا اسکرپٹ ورژن	ویب براؤزر
JavaScript 1.0	Navigator 2.0, Internet Explorer 3.0
JavaScript 1.3	Navigator 4.06-4.7x, Internet Explorer 4.0
JavaScript 1.5	Navigator 6.0, Mozilla, Internet Explorer 5.5 onwards
JavaScript 1.6	Mozilla Firefox 1.5
JavaScript 1.7	Mozilla Firefox 2.0
JavaScript 1.8	Mozilla Firefox 3.0

مائکروسافٹ نے جاوا اسکرپٹ کے کئی ورژن جاری کیے ہیں۔ فی الحال انٹرنیٹ ایکسپلورر 7.0 کے ساتھ جاوا اسکرپٹ کا ورژن 15.7 استعمال ہو رہا ہے۔

10.2 کلائنٹ سرور ماڈل

آگے بڑھنے سے پہلے ہمیں نوڈ (Node)، کلائنٹ (Client) اور سرور (Server) کے بارے میں جان لینا چاہیے۔

10.2.1 نوڈ

نوڈ ایک ایسا ٹرمینل ہے جو نیٹ ورک سے منسلک رہتا ہے۔ لیپ ٹاپ، PDAs، انٹرنیٹ آراسٹہ موبائل وغیرہ کو کمپیوٹر نیٹ ورک میں ایک نوڈ کے طور پر سمجھا جاسکتا ہے۔

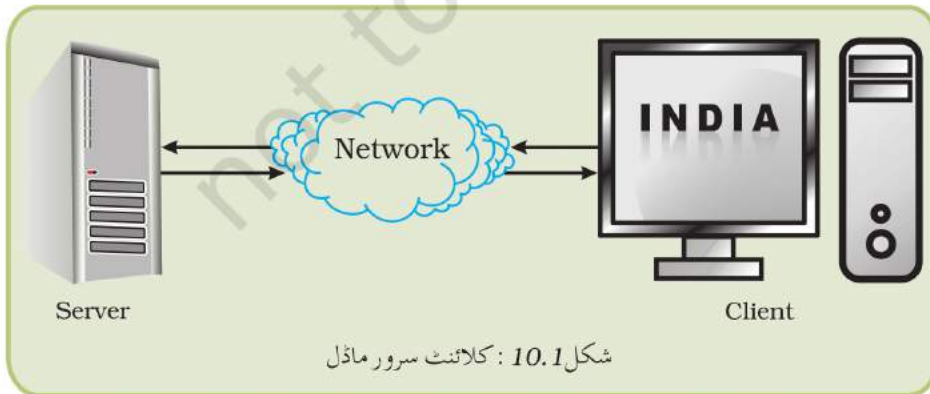
10.2.2 کلائنٹ

یہ ایک نوڈ کمپیوٹر ہے جو سرور کے ساتھ جڑ جاتا ہے، استعمال کنندہ سے ڈیٹا کو جمع کرتا ہے، اسے سرور کو بھیجتا ہے، سرور سے معلومات حاصل کرتا ہے اور اسے استعمال کنندہ کے سامنے پیش کر دیتا ہے۔

10.2.3 سرور

کلائنٹ سرور ماڈل کے تناظر میں، سرور کلائنٹ کا ہم منصب ہے۔ یہ ایک ایسا کمپیوٹر ہے جو کلائنٹ کی دریافت (پوچھتاچھ) پر عمل درآمد کرتا ہے۔ وہ پروگرام جو کلائنٹ کی درخواست کے تینوں ردعمل ظاہر کرتے ہیں سرور اپیلی کیشن کہلاتے ہیں۔ وہ کمپیوٹر جن پر سرور اپیلی کیشن کو چلایا جاتا ہے سرور مشین کہلاتے ہیں۔ ویب سرور، ڈیٹا بیس سرور (Database server) اور میل سرور، سرور کی چند مثالیں ہیں۔

کلائنٹ سرور ماڈل ایسا کمپیوٹر نیٹ ورک ہے جہاں کلائنٹ اور سرور نیٹ ورک کے ذریعہ ایک دوسرے سے تفاعل کرتے ہیں (شکل 10.1)۔ کلائنٹ استعمال کنندہ سے ان پٹ کے طور پر ڈیٹا جمع کرتا ہے اور سرور کو درخواست بھیجتا ہے۔ سرور درخواست کی پروسسنگ کرتا ہے اور متعلقہ معلومات کلائنٹ کو بھیج دیتا ہے۔ ریلوے ریزرویشن سسٹم، آن لائن بینکنگ اور آن لائن گیمنگ کلائنٹ سرور کی مثالیں ہیں۔



جاوا اسکرپٹ کی مدد سے کلائنٹ سائڈ اسکرپٹنگ

10.2.4 کلائنٹ - سائڈ جاوا اسکرپٹ

کلائنٹ - سائڈ جاوا اسکرپٹ سے مراد وہ جاوا اسکرپٹ ہے جس پر ویب براؤزر کے ذریعہ کلائنٹ مشین پر عمل درآمد کیا جاتا ہے۔ جاوا اسکرپٹ اسٹیمنٹ HTML دستاویز میں داخل کیے جاسکتے ہیں جو کسی میٹ ورک کنکشن کے بغیر مائکس کلک، فارم ان پٹ اور پیج نیویگیشن جیسے واقعوں کے تئیں رد عمل کر سکتے ہیں۔

10.2.5 سرور - سائڈ جاوا اسکرپٹ

سرور - سائڈ جاوا اسکرپٹ کو "LiveWire" بھی کہا جاتا ہے۔ کلائنٹ - سائڈ جاوا اسکرپٹ کی طرح سرور - سائڈ جاوا اسکرپٹ کو بھی HTML دستاویز میں داخل کیا جاسکتا ہے۔ جب کلائنٹ کے ذریعہ پوچھتاچھ (Query) کی جاتی ہے تو ویب سرور اسکرپٹ کی ترجمانی کرنے کے بعد اس پر عمل درآمد کرتا ہے۔

10.3 جاوا اسکرپٹ کے ساتھ کام کی شروعات

جاوا اسکرپٹ ایک مکمل لینگویج نہیں ہے اور اسے HTML دستاویز میں داخل کرنا پڑتا ہے۔ یا پھر کسی ایسی بیرونی فائل کا حوالہ دیا جاتا ہے جس میں جاوا اسکرپٹ کوڈ موجود ہوتے ہیں۔ ہم لفظ 'اسکرپٹ' یا 'پروگرام' کا استعمال جاوا اسکرپٹ کوڈ کے لیے کر سکتے ہیں۔ کسی اسکرپٹ کو استعمال کرنے کا سب سے عام طریقہ یہ ہے کہ HTML دستاویز میں `<script>` HTML اور `</script>` ٹیگ کا استعمال کیا جاتا ہے۔ ہم اپنے جاوا اسکرپٹ کوڈ کو HTML دستاویز کے HEAD یا BODY سیکشن میں داخل کر سکتے ہیں۔

سنٹیکس (عمومی فارمولا) مندرجہ ذیل ہے۔

جاوا اسکرپٹ کوڈ کی ابتدا کو ظاہر کرتا ہے

.....JavaScript statement(s);

</SCRIPT>

جاوا اسکرپٹ کوڈ کے اختتام کو ظاہر کرتا ہے

مندرجہ ذیل جدول میں اسکرپٹ ایٹری بیوٹ (Script attributes)، قدریں اور ان کی تفصیل دی گئی ہے۔

ایٹری بیوٹ (Attribute)	قدر (Value)	تفصیل (Description)
Type	text/javascript text/ecmascript text/vbscript	اسکرپٹ کی قسم
Language	Javascript vbscript	اسکرپٹنگ لینگویج کا نام
Src	URL	اسکرپٹ پر مشتمل فائل کا URL

پروگرام 10.1: document.write() کا استعمال کر کے ابتدائی سادہ جاوا اسکریپٹ پروگرام۔



نوٹ: آپ نے دیکھا کہ کوڈ ایک سطر میں نہیں آ سکا ہے لہذا ہم نے کوڈ کے تسلسل کو ظاہر کرنے کے لیے 8 کا استعمال کیا ہے۔ کوڈ لکھتے وقت آپ کو اسے ٹائپ کرنے کی ضرورت نہیں ہے۔

مذکورہ بالا پروگرام کو چلانے کے لیے کسی بھی ٹیکسٹ ایڈیٹر مثلاً نوٹ پیڈ، ورڈ پیڈ کا استعمال کر کے مذکورہ بالا پروگرام کوڈ ٹائپ کیجیے اور اسے "filename.html" (مثلاً abc.html) کے طور پر محفوظ کیجیے۔ کسی بھی براؤزر اپیلی کیشن (مثلاً انٹرنیٹ ایکسپلورر، موزیلا فائر فوکس، اوپیرا وغیرہ) کے ذریعہ اس فائل کو کھولیے۔

جاوا اسکریپٹ کوڈ کو لکھنے اور چلانے کے لیے درکار ٹولز

جاوا اسکریپٹ کوڈ کو استعمال کرنے کے لیے مندرجہ ذیل ٹولز کی ضرورت ہوتی ہے۔

- (a) ٹیکسٹ ایڈیٹر: ہم کسی بھی ٹیکسٹ ایڈیٹر یا ورڈ پروسیسر (مثلاً نوٹ پیڈ، ورڈ پیڈ وغیرہ) کا استعمال کر سکتے ہیں۔
- (b) براؤزر: براؤزر جاوا اسکریپٹ کوڈ کی ترجمانی کرتا ہے اور آؤٹ پٹ کو براؤزر کی ڈاکیومنٹ ونڈو میں ظاہر کر دیتا ہے۔

10.3.1 جاوا اسکریپٹ کوڈ کو متعین کرنا

جاوا اسکریپٹ کوڈ کو متعین کرنے کے دو طریقے ہیں:

- 1- داخلی (پوسٹ) / ان لائن جاوا اسکریپٹ: جاوا اسکریپٹ کوڈ کو HTML دستاویز کے HEAD یا BODY سیکشن میں متعین کیا جاسکتا ہے۔

(a) اگر جاوا اسکریپٹ کوڈ کو ایک سے زیادہ مرتبہ استعمال کرنا مقصود ہو تو اسے HEAD سیکشن میں متعین کرنے کی صلاح دی جاتی ہے۔

(b) اگر جاوا اسکریپٹ کوڈ مختصر ہے اور اسے صرف ایک مرتبہ استعمال کیا جانا ہے تو اسے HTML دستاویز کے BODY سیکشن میں متعین کرنے کی صلاح دی جاتی ہے۔

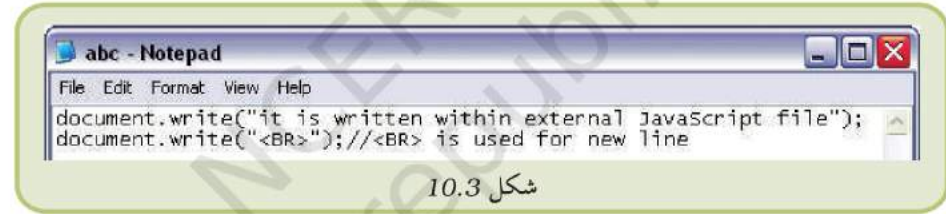
جاوا اسکرپٹ کی مدد سے کلائنٹ سائڈ اسکرپٹنگ

2۔ بیرونی جاوا اسکرپٹ: اگر ایک ہی جاوا اسکرپٹ کوڈ کو متعدد دستاویزات میں استعمال کیا جانا ہے تو اس کا سب سے عمدہ طریقہ یہ ہے کہ جاوا اسکرپٹ کوڈ کو ایک علاحدہ فائل (بیرونی فائل) میں متعین کیا جائے جس کا ایکسٹینشن "js" ہو۔ اس کام کو انجام دینے کے لیے ہم <SCRIPT> ٹیگ میں src ایٹریبیوٹ کا استعمال کریں گے جو جاوا اسکرپٹ فائل سے اس کے تعلق (Link) کو ظاہر کرے گا۔

مثال: بیرونی جاوا اسکرپٹ کوڈ کے استعمال کی مثال۔

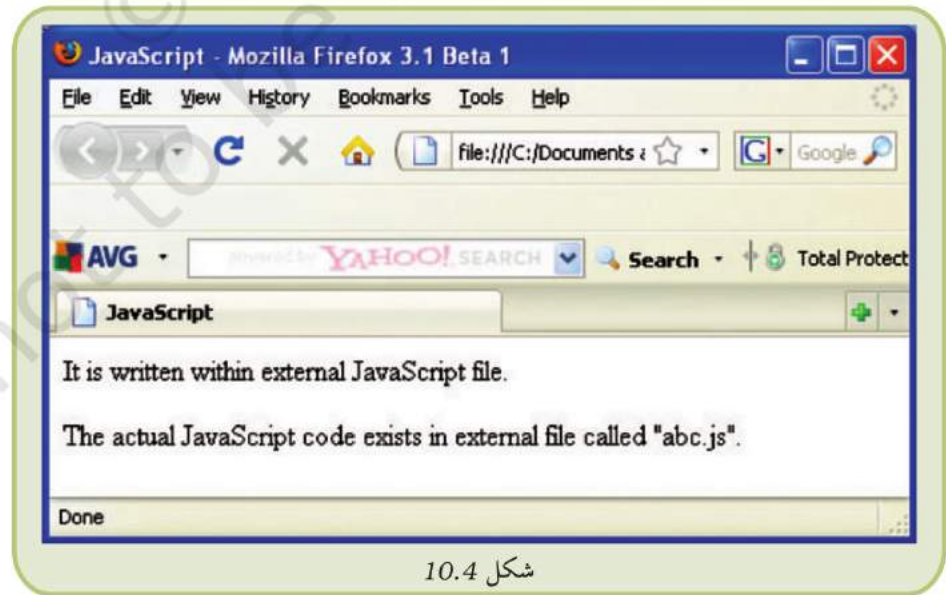
```
<HTML>
<HEAD>
<TITLE>Using External JavaScript</TITLE>
</HEAD>
<BODY>
<SCRIPT language="JavaScript" src="abc.js">
</SCRIPT>
<P> The actual JavaScript code exists in external file called
"abc.js". </P>
</BODY>
</HTML>
```

حقیقی جاوا اسکرپٹ فائل "abc.js" ہے۔



شکل 10.3

آؤٹ پٹ



شکل 10.4

10.4 جاوا اسکرپٹ میں اسٹیٹمنٹ

اسٹیٹمنٹ وہ کمانڈ یا ہدایت ہیں جو مجوزہ کاموں کو انجام دینے کے لیے جاوا اسکرپٹ شارح (Interpreter) کو دی جاتی ہیں۔ جاوا اسکرپٹ انٹرپرائز تقریباً سبھی انٹرنیٹ براؤزر کے اندر موجود ہوتا ہے۔ کسی کام کو انجام دینے کے لیے متعین کیے گئے اسٹیٹمنٹ کا مجموعہ اسکرپٹ یا پروگرام کہلاتا ہے۔ جاوا اسکرپٹ اسٹیٹمنٹ مندرجہ ذیل کی طرح نظر آئے گا:

```
a = 100;           // stores value 100 in variable a
b = 200;           // stores value 200 in variable b
c = a + b;         // stores the sum of a and b in
                   // variable c

document.write
("Sum of A and B : "); // displays the string
document.write(c);     // displays the value of c
```

جاوا اسکرپٹ میں ";" (Semicolon) کا استعمال اسٹیٹمنٹ کے اختتام کے لیے کیا جاتا ہے۔ لیکن، اگر دو اسٹیٹمنٹ علاحدہ علاحدہ لائنوں میں لکھے جاتے ہیں تو ہر کوئی (:) کا استعمال نہیں ہوگا۔ چند موزوں (درست) اسٹیٹمنٹ ذیل میں دیے گئے ہیں:

- (i) p=10
q=20
- (ii) x=12; y=25 // semicolon(:) separating two statements.

چند غیر موزوں (غلط) اسٹیٹمنٹ:

x=12 y=25 // statements within the same line not separated by semicolon (:)

10.4.1 تبصرے

تبصرے (comments) ایسے بیانات (Statements) ہیں جنہیں انٹرپرائز ہمیشہ نظر انداز کر دیتا ہے۔ ان کا استعمال اسٹیٹمنٹ پر تبصرہ کرنے کے لیے کیا جاتا ہے تاکہ دیگر پروگرامر ان اسٹیٹمنٹ کو آسانی سے پڑھ اور سمجھ سکیں۔

- ایک سطر میں کیا گیا تبصرہ جس میں دوہرے سلیش (/ /) کا استعمال ہوتا ہے۔
- کئی سطروں میں کیا گیا تبصرہ جس میں /* اور */ کا استعمال ہوتا ہے۔

مثال کے طور پر

```
// This is a single-line comment.

/* This is a multiple-line comment.

It can be of any length. */
```

10.4.2 لٹرل

لٹرل (Literals) وہ مستقل قدریں ہیں جنہیں جاوا اسکرپٹ کوڈ میں براہ راست استعمال کیا جاتا ہے۔ مثال کے طور پر:

```
a=10;
```

```
b=5.7;
```

```
document.write("Welcome");
```

مذکورہ بالا اسٹیٹمنٹ میں 10، 5.7، "Welcome" لٹرل (literal) ہیں۔

10.4.3 شناخت کار

شناخت کار (Identifier) پروگرامر کے ذریعہ تشکیل دیے گئے متغیرات (Variables)، فنکشن (Function)، array وغیرہ کے نام ہیں۔ یہ چھوٹے اور بڑے حروف کا کوئی بھی سلسلہ ہو سکتا ہے، جس میں اعداد، انڈر اسکور () اور ڈالر کا نشان (\$) بھی شامل ہیں۔ شناخت کار کے شروع میں کوئی عدد نہیں ہونا چاہیے اور اس کا نام جاوا اسکرپٹ کے کلیدی الفاظ کے مشابہ نہیں ہونا چاہیے۔

چند موزوں (درست) اور غیر موزوں (غلط) شناخت کار مندرجہ ذیل ہیں:

موزوں:

```
RollNo
```

```
bus_fee
```

```
_vp
```

```
$amt
```

غیر موزوں:

```
to day // Space is not allowed
```

```
17nov // must not begin with a number
```

```
%age // no special character is allowed
```

10.4.4 محفوظ الفاظ یا کلیدی الفاظ

محفوظ الفاظ کا استعمال جاوا اسکرپٹ انٹرپرائز کو ہدایت دینے کے لیے کیا جاتا ہے اور ہر ایک محفوظ لفظ کے مخصوص معنی ہوتے ہیں۔ انہیں پروگرام میں آڈینیٹی فائر کے طور پر استعمال نہیں کیا جاسکتا۔ اس کا مطلب یہ ہے کہ ہم محفوظ الفاظ کا استعمال متغیرات، arrays، آبجیکٹ، فنکشن وغیرہ کا نام رکھنے کے لیے نہیں کر سکتے۔ ان الفاظ کو "کلیدی الفاظ" بھی کہا جاتا ہے۔ جاوا اسکرپٹ کے محفوظ الفاظ کی فہرست ضمیمہ 10.1 میں دی گئی ہے۔

10.4.5 متغیرات

متغیرہ ایک ایسا آڈینٹی فائر ہے جسے قدریں تفویض کی جاسکتی ہیں۔ اسکرپٹ نافذ ہونے کے دوران ان قدروں کو تبدیل کیا جاسکتا ہے۔ کسی متغیرہ میں قدر اسٹور ہو جانے کے بعد اسے متغیرہ کے نام کا استعمال کر کے ایکسس کیا جاسکتا ہے۔ متغیرات کو بیان (Declare) کرنا لازمی نہیں ہے حالانکہ ایسا کرنا اچھی عادت میں شمار ہوتا ہے۔ متغیرہ کو بیان کرنے کے لیے کلیدی لفظ **var** کا استعمال کیا جاتا ہے۔

سنٹیکس

var var-name [= value] [... , var-name [= value]]

مثال

```
var name = "Sachin";           // Here 'name' is variable
document.write(name);         // Prints Sachin
```

جاوا اسکرپٹ متغیرہ کسی بھی ڈیٹا ٹائپ کا متحمل ہو سکتا ہے۔ مثال کے طور پر:

```
i = 7;
document.write(i);           // prints 7
i = "seven";                 // JavaScript allows to assign string values
document.write(i);           // prints seven
```

متغیرہ کو بیان کرنے کی چند موزوں مثالیں:

```
var cost;
var num, cust_no = 0;
var amount = 2000;
```

نام رکھنے کا طریقہ کار

ہمیں متغیرہ کے با معنی نام رکھنے چاہئیں۔ متغیرہ کا نام حرف، انڈر اسکور () یا ڈالر کے نشان (\$) سے شروع ہونا چاہیے۔ اس کے بعد کی علامتیں ہندسے (0-9) ہو سکتے ہیں۔ جاوا اسکرپٹ میں چھوٹے اور بڑے حروف کے علاوہ علاحدہ معنی ہیں لہذا my_school متغیرہ نام اور My_School متغیرہ یکساں نہیں ہیں۔

درست (موزوں) متغیرہ ناموں کی مثالیں

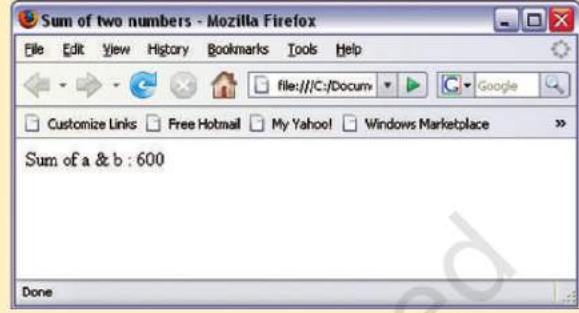
```
f_name
India123
_sumof
```

غیر موزوں متغیرہ ناموں کی مثالیں

```
10_numbers - must not begin with any number.
rate% - '%' is not a valid character.
my name - Space is not allowed.
```

پروگرام 10.2: var کا استعمال کر کے دو اعداد کا حاصل جمع معلوم کرنا

```
<HTML>
<HEAD>
<TITLE>Sum of two numbers</TITLE>
</HEAD>
<BODY>
<SCRIPT type="text/javascript">
var a = 100;
var b = 500;
var c = a + b;
document.write ("Sum of a & b : "
: " + c );
</SCRIPT>
</BODY>
</HTML>
```



شکل 10.5

10.5 ڈیٹا ٹائپ

جاوا اسکریپٹ میں تین بنیادی ڈیٹا ٹائپ - اسٹرنگ (String)، بولین (Boolean) اور دو مرکب ڈیٹا ٹائپ - arrays اور آجیکٹ کا استعمال ہوتا ہے۔

10.5.1 عدد

عدد متغیر صحیح عدد یا حقیقی عدد کسی بھی قسم کے عدد پر مشتمل ہو سکتا ہے۔ اعداد کی چند مثالیں مندرجہ ذیل ہیں:

29, -43, 3.40, 3.4323

10.5.2 اسٹرنگ

اسٹرنگ (String) حروف، ہندسوں، رموز اوقاف (Punctuation)، حرفی علامات (Characters) وغیرہ کا مجموعہ ہے۔ اسٹرنگ کی علامتوں کو واحد واوین یا دوہرے واوین (’ یا ’’) کے اندر لکھا جاتا ہے۔ اسٹرنگ کی مثالیں مندرجہ ذیل ہیں:

‘welcome’, “7.86”, “wouldn’t you exit now”, ‘country=’India’

جاوا اسکریپٹ ہمیں اسٹرنگ کے اندر Escape sequence کے استعمال کی سہولت فراہم کرتی ہے۔ Escape sequence بیک سلیش (\) کے ساتھ شروع ہوتی ہے اور اس کے بعد دیگر حرفی علامات کو لکھا جاتا ہے۔ بیک سلیش براؤزر کو ایک مخصوص عمل یا حرفی علامت کو ظاہر کرنے کی ہدایت دیتا ہے۔ مثال کے طور پر “\” ایک Escape sequence ہے جو دوہرے واوین (”) کو ظاہر کرتی ہے۔

Escape Sequence	Action/ Character Represented
\b	Backspace
\n	New line
\r	Carriage return
\t	Tab
\'	Single quote (')
\"	Double quote (")
\\	Backslash (\)

مثال :

```
document.write ("Abhinav said, \"Earth doesn't revolve round the sun\". But teacher corrected him.");
```

یہاں اس مثال میں دو قسم کے ایسکیپ کیڑے استعمال کیا گیا ہے یعنی \" اور \\'۔

نتیجہ

Abhinav said, "Earth doesn't revolve round the sun". But teacher corrected him.

10.5.3 بولین قدریں (Boolean Values)

بولین متغیرہ میں دو ممکنہ قدریں ہیں۔ صحیح (true) یا غلط (false) اسٹوریج کی جاسکتی ہیں۔ داخلی طور پر true کے لیے 1 اور false کے لیے 0 کو اسٹوریج کیا جاتا ہے۔ اسے شرائط کا نتیجہ (output) حاصل کرنے کے لیے استعمال کیا جاتا ہے کہ آیا شرط کا نتیجہ صحیح ہے یا غلط۔

مثال

```
x== 100; // results true if x=100 otherwise false.
```

Arrays 10.5.4

array ایک ہی قسم کی مشترک نام والی ڈیٹا قدروں کا مجموعہ ہے۔ array کے ہر ایک ڈیٹا ایلیمنٹ کا array میں اس کے مقام کے ذریعہ حوالہ دیا جاتا ہے جسے اس کا انڈیکس نمبر بھی کہا جاتا ہے۔ انفرادی array ایلیمنٹ (عنصر) کا حوالہ array کا نام اور اس کے بعد مربع بریکٹ میں انڈیکس نمبر لکھ کر دیا جاسکتا ہے۔ جاوا اسکرپٹ میں انڈیکس نمبر صفر سے شروع ہوتا ہے یعنی جاوا اسکرپٹ میں پہلے عنصر کی انڈیکس قدر 0 اور دوسرے عنصر کی 1 ہوتی ہے۔ اسی طرح یہ سلسلہ آگے کی طرف بڑھتا جاتا ہے۔ array کو مندرجہ ذیل کسی بھی طریقہ سے ظاہر کیا جاسکتا ہے :

```
var a = new a();
var x = [];
var m = h[2,4,"sun"];
```

Array کی شروعات متعینہ قدروں کے ساتھ ہوتی ہے کیوں کہ اس کے عناصر اور اس کی لمبائی متعینہ آرگیومنٹ (Argument) کی تعداد سے طے ہوتی ہے۔

مثال: یہ تین عناصر پر مشتمل array کی تشکیل کرتا ہے جس کا نام games ہے۔

```
games = ["Hockey", "Cricket", "Football"];
```

ہم array میں قدروں کی مختلف اقسام بھی اسٹور کر سکتے ہیں۔

مثال کے طور پر:

```
var arr = new Array();           // creation of an array
arr[0] = "JAVASCRIPT";          // stores String literal at index 0
arr[1] = 49.5;                  // stores real number at index 1
arr[2] = true;                  // stores Boolean value
```

10.5.5 معدوم قدر

جاوا اسکریپٹ میں ایک مخصوص ڈیٹا ٹائپ کو بھی استعمال کیا جاسکتا ہے، جسے **null** کہا جاتا ہے۔ یہ "no value or blank" کو ظاہر کرتا ہے۔ نوٹ کیجیے کہ **null** صفر کے مساوی نہیں ہے۔

مثال:

```
var distance = new object();
distance = null;
```

10.6 آبجیکٹ

جاوا اسکریپٹ آبجیکٹ پر مبنی اسکریپٹنگ لینگویج ہے۔ اس کی مدد سے ہم اپنے آبجیکٹ متعین کر سکتے ہیں اور اپنے متغیر بنا سکتے ہیں۔ یہ پہلے سے متعین آبجیکٹ کا مجموعہ بھی پیش کرتی ہے۔ ہمارے ویب صفحہ پر موجود جدولیں، فارم، بٹن، امیج (Image) یا لنک (Link) آبجیکٹ کی مثالیں ہیں۔ آبجیکٹ سے وابستہ قدریں اس کی خاصیتیں (Properties) ہوتی ہیں اور ان کے ذریعہ آبجیکٹ پر انجام دیے جانے والے عمل میٹھڈس (Methods) یا طرز عمل (Behaviour) کہلاتے ہیں۔ آبجیکٹ سے متعلق خاصیت کو مندرجہ ذیل طریقہ سے دیکھا جاسکتا ہے:

```
ObjectName.PropertyName
```

اب ہم جاوا اسکریپٹ میں پہلے سے متعین آبجیکٹ کا مطالعہ کریں گے۔

10.6.1 ڈاکیومنٹ آبجیکٹ

ڈاکیومنٹ آبجیکٹ ونڈو آبجیکٹ کا ایک حصہ ہے۔ **window.document** پراپرٹی کے ذریعہ اس تک رسائی کی جاسکتی ہے۔ **document** آبجیکٹ HTML دستاویز کو ظاہر کرتا ہے اور یہ HTML دستاویز میں موجود کئی عناصر تک رسائی کی اجازت دیتا ہے۔ مثال کے طور پر موجودہ دستاویز کے عنوان (Title) کو **document.title** پراپرٹی کے ذریعہ دیکھا جاسکتا ہے۔

ڈاکیومنٹ آبجیکٹ کی عام خاصیتیں:

پراپرٹیز	مقاصد
Title	returns/ sets title of the current document
bgColor	returns/ sets title of the current document
fgColor	returns/ sets the text color of the current document
linkColor	returns/ sets the color of hyperlinks in the document
alinkColor	returns/ sets the color of active links in the document
vlinkColor	returns/ sets the color of visited hyperlinks.
height	returns the height of the current document
width	returns the width of the current document
Forms	returns a list of the FORM elements within the current document
Images	returns a list of the images in the current document
URL	returns a string containing the URL of the current document
Location	to load another URL in current document window

Methods

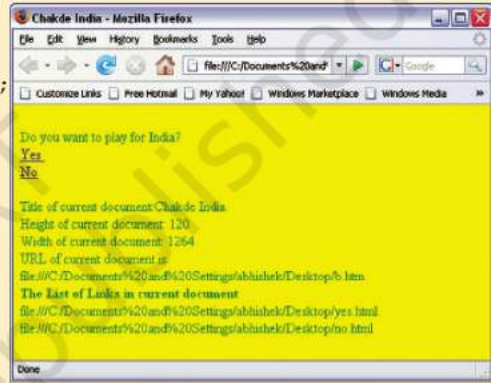
Purposes

open()	Opens a document for writing.
write()	Writes string/data to a document.
writeln()	Writes string/data followed by a newline character to a document.
close()	Closes a document stream for writing.

پروگرام 10.3 : ڈاکیومنٹ آبجیکٹ کی خاصیتیں اور طرز عمل کی مثالیں

```
<HTML>
<HEAD>
<TITLE>Document Properties</TITLE>
</HEAD>
<BODY>
<SCRIPT type="text/javascript">
```

```
document.fgColor = "green"; // sets text color
document.bgColor = "yellow"; // background color
document.title = "Chakde India"; // change title
document.linkColor = "navy"; // hyperlinks color
document.alinkColor = "red"; // active links
document.vlinkColor = "lime"; // visited hyperlinks
document.write("<BR>Do you want to play for India?");
document.write("<BR> <A href='yes.html'> Yes </A>");
document.writeln("<BR> <A href='no.html'>No</A><BR>");
document.write("<BR>Title of current document: " + document.title);
document.write("<BR>Height of current document: " + document.height);
document.write("<BR>Width of current document: " + document.width);
document.write("<BR> URL of current document is: " + document.URL);
//Use of document.links to list of all the hyperlinks
document.write("<BR><B>The List of Links in current document</B>");
var links = document.links;
for(var i = 0; i < links.length; i++)
{
document.write("<BR>"+ document.links[i]);
}
</SCRIPT>
</BODY>
</HTML>
```



شکل 10.6

10.6.2 ڈیٹ آ بجیکٹ

اس آ بجیکٹ کا استعمال تاریخ کو متعین کرنے اور اس میں ردوبدل کرنے کے لیے کیا جاتا ہے۔ جاوا اسکرپٹ تاریخیں January 1, 1970 آدھی رات سے ملی سیکنڈ کی شکل میں اسٹور کی جاتی ہیں۔ یہ تاریخ عہد (Epoch) کہلاتی ہے۔ 1970 سے پہلے کی تاریخیں منفی اعداد کے ذریعہ ظاہر کی جاتی ہیں۔ date آ بجیکٹ کی تشکیل Date() کے ساتھ نئے کلیدی الفاظ کی مدد کی جاسکتی ہے۔

سینٹیکس

```
new Date()
new Date(milliseconds)
new Date(dateString)
new Date(yr_num, mo_num, day_num
[, hr_num, min_num, sec_num, ms_num])
```

Parameters

Milliseconds	Milliseconds since 1 January 1970 00:00:00.
dateString	Date String, e.g. "October 5, 2007"
yr_num, mo_num, day_num	Year (e.g. 2007) Month (Value 0-11, 0 for January and 11 for December), Day (1-31)
hr_num, min_num, sec_num, ms_num	Values for Hour, Minutes, Second and milliseconds

date() کے استعمال کی مختلف مثالیں

```
today = new Date();
dob = new Date("October 5, 2007 12:50:00");
doj = new Date(2007, 10, 5);
bday = new Date(2007, 10, 5, 12, 50, 0);
```

تاریخ کی قدروں (date values) کو پڑھنے کے طریقے

ہم Date آبجیکٹ سے قدروں کو حاصل کرنے کے لیے get methods کا استعمال کر سکتے ہیں۔ یہاں کچھ get methods دیے گئے ہیں جو مقامی وقت کے اعتبار سے قدروں کو ظاہر کرتے ہیں:

getDate()	مہینہ کا دن ظاہر کرتا ہے
getDay()	ہفتہ کا دن ظاہر کرتا ہے
getFullYear()	مکمل سال ظاہر کرتا ہے
getHours()	گھنٹہ ظاہر کرتا ہے
getMinutes()	منٹ کو ظاہر کرتا ہے
getMonth()	مہینہ کو ظاہر کرتا ہے
getSeconds()	سیکنڈ کو ظاہر کرتا ہے
getTime()	وقت کی عددی قدر کو ظاہر کرتا ہے
getYear()	سال کو ظاہر کرتا ہے

پروگرام 10.4: ایک سادہ جاوا سکرپٹ پروگرام جو آج کی تاریخ اور موجودہ وقت کو ظاہر کرتا ہے۔

```
<HTML>
<HEAD>
  <TITLE>Displaying Time</TITLE>
</HEAD>
<BODY>
  <CENTER>
```

```
<H1>Today's Date and Current Time</H1>
</CENTER>
<SCRIPT type="text/javascript">
var today = new Date();
document.write("<H2>"); // JavaScript allows the use
document.write(today); // of HTML formatting tag
document.write("</H2>"); // with document.write
</SCRIPT>
</BODY>
</HTML>
```



شکل 10.7

Math 10.6.3 آبجیکٹ

یہ آبجیکٹ زیادہ پیچیدہ ریاضیاتی اعمال کو انجام دینے کے لیے میتھڈس (Methods) اور مستقلوں (Constants) پر مشتمل ہوتا ہے۔ یہ آبجیکٹ اتنی تیزی سے کام نہیں کر سکتا جتنی تیزی سے دیگر آبجیکٹ کر سکتے ہیں۔ **Math** کی سبھی خاصیتیں (Properties) اور میتھڈس سکونی (غیر متحرک) ہوتے ہیں۔ ہم مستقلہ **Math.PI** کو **p** سے ظاہر کر سکتے ہیں اور **sine** فنکشن کو **Math.sin(x)** کے ذریعہ، جہاں **x** میتھڈ کا آرگیومنٹ (Argument) ہے۔

پراپریٹیز	تفصیل
Math.PI	p کی قدر کو ظاہر کرتا ہے
Math.E	یولر (Euler) کا مستقلہ اور فطری لوگارتم (Natural Logarithm)
Math.LN2	2 کا فطری لوگارتم
Math.LN10	10 کا فطری لوگارتم تقریباً 2.302
SQRT1_2	1/2 کا جذر المربع
SQRT2	2 کا جذر المربع
میتھڈس	تفصیل
pow(x, p)	X ^p کو ظاہر کرتا ہے۔
abs(x)	x کی مطلق قدر کو ظاہر کرتا ہے۔
exp(x)	e ^x کو ظاہر کرتا ہے۔
log(x)	x کے فطری لوگارتم کو ظاہر کرتا ہے۔

x کے جذر المربع کو ظاہر کرتا ہے۔	sqrt(x)
0 اور 1 کے درمیان بے ترتیب (Random) اعداد کو ظاہر کرتا ہے۔	random()
x سے بڑے یا اس کے مساوی سب سے چھوٹے صحیح عدد کو ظاہر کرتا ہے۔	ceil(x)
x سے چھوٹے یا اس کے مساوی سب سے بڑے صحیح عدد کو ظاہر کرتا ہے۔	floor(x)
x اور y میں سے چھوٹے عدد کو ظاہر کرتا ہے۔	min(x, y)
x اور y میں سے بڑے عدد کو ظاہر کرتا ہے۔	max(x, y)
x کو قریب ترین صحیح عدد تک راؤنڈ اپ یا ڈاؤن کرتا ہے۔	round(x)
x کی sin قدر کو ظاہر کرتا ہے، جہاں x ریڈین میں ہے۔	sin(x)
x کی cos قدر کو ظاہر کرتا ہے، جہاں x ریڈین میں ہے۔	cos(x)
x کی tan قدر کو ظاہر کرتا ہے، جہاں x ریڈین میں ہے۔	tan(x)

مثال: Math آبجیکٹ کی پراپرٹیز اور میٹھڈس کی وضاحت کرنا۔

```
<HTML>
<HEAD>
<TITLE>Math Object</TITLE>
</HEAD>
<BODY>
<SCRIPT type="text/JavaScript">
document.write("Value of PI :"+Math.PI + "<BR>");
document.write("Random value:"+Math.random()+"<BR>");
document.write("Rounded value of 0.69 :"+
Math.round(0.69)+"<br>");
document.write("Value of 5 <SUP>2</SUP> :"+
Math.pow(5,2) + "<br>");
document.write("Square root of 2 :"+Math.SQRT2 );
</SCRIPT>
</BODY>
</HTML>
```



شکل 10.8

10.7 عباراتیں (Expressions) اور آپریٹر

عبارت (Expression) ایسے آپریٹر اور آپرینڈ (Operand) کا مجموعہ ہے جن کی تحسیب کی جاسکتی ہے۔ اس میں فنکشن کال بھی شامل ہو سکتی ہیں جو قدروں کو ظاہر کرتی ہیں۔

مثالیں

```
x = 7.5 // a numeric literal
"Hello India!" // a string literal
false // a Boolean literal
{feet:10, inches:5} // an object literal
[2,5,6,3,5,7] // an array literal
v = m + n; // the variable v
tot // the variable tot
```

10.7.1 اریٹھمٹک آپریٹر

ان کا استعمال حسابی / ریاضیاتی اعمال کو انجام دینے کے لیے کیا جاتا ہے مثلاً جمع، تفریق (گھٹانا)، ضرب، تقسیم وغیرہ۔ اریٹھمٹک آپریٹر ایک یا ایک سے زیادہ عددی قدروں (حرفی علامتیں یا متغیر) پر کام کرتے ہیں اور واحد عددی قدر کو ظاہر کرتے ہیں۔ بنیادی اریٹھمٹک آپریٹر مندرجہ ذیل ہیں:

+	(جمع)	-	(تفریق/گھٹانا)
*	(ضرب)	/	(تقسیم)
%	(ماڈیولس)	++	(1 کا اضافہ)
--	(1 کم کرنا)		

مثال

```
var s = 10 + 20; // result: s=30
var h = 50 * 4; // result: h = 200
var d = 100 / 4; // result: d = 25
var r = 72 % 14; // result: r=2
```

اضافہ یا کم کرنے والے آپریٹر

ان آپریٹر کا استعمال متغیر کی قدر میں 1 کا اضافہ کرنے یا اس میں 1 کی کمی لانے کے لیے کیا جاتا ہے۔ ان آپریٹروں کا استعمال کر کے تحسیب کا عمل تیزی سے انجام دیا جاسکتا ہے۔

مثال

```
var a = 15; // result: a = 16
a++;
var b = 20; // result: b = 19
b--;
```

10.7.2 تفویض (Assignment) آپریٹر

یہ اپنے دائیں آپرینڈ کی قدر بائیں آپرینڈ کو تفویض کر دیتا ہے۔ اس آپریٹر کو مساوی نشان (=) کے ذریعہ ظاہر کیا جاتا ہے۔

مثال

// یہ بیان x کو قدر 100 تفویض کر دیتا ہے

x = 100

جاوا اسکریپٹ میں معیاری اعمال کے لیے شارٹ ہینڈ آپریٹر کا بھی استعمال کیا جاسکتا ہے۔

شارٹ ہینڈ آپریٹر	مثال	کے معادل
+	a + = b	a = a + b
-	a - = b	a = a - b
*	a * = b	a = a * b
/	a / = b	a = a / b
%	a % = b	a = a % b

10.7.3 تقابلی آپریٹر

تقابلی آپریٹر ایسی علامتیں ہیں جو شرط کی تحسیب کرنے کے بعد یہ ظاہر کرتی ہیں کہ بولین قدر درست (True) ہے یا غلط (False)۔

مثال کے طور پر x > y سے جو درست نتیجہ حاصل ہوگا وہ یہ کہ متغیر x متغیر y سے بڑا ہے۔

بیک جاوا اسکریپٹ تقابلی آپریٹر مندرجہ ذیل جدول میں دیے گئے ہیں:

آپریٹر	تفصیل	مثال
=	کے مساوی ہے	4 == 8 کو غلط (False) ظاہر کرتا ہے۔
!=	کے مساوی نہیں ہے	4 != 8 کو درست (True) ظاہر کرتا ہے۔
>	سے بڑا ہے	8 > 4 کو درست (True) ظاہر کرتا ہے۔
<	سے چھوٹا ہے	8 < 4 کو غلط (False) ظاہر کرتا ہے۔
<=	سے چھوٹا ہے یا مساوی ہے	8 <= 4 کو غلط (False) ظاہر کرتا ہے۔
>=	سے بڑا ہے یا مساوی ہے	8 >= 4 کو درست (True) ظاہر کرتا ہے۔

تقابلی آپریٹر اسٹرنگ کے لیے بھی کام کرتے ہیں۔ یہاں موازنہ حروف تہجی کے اعتبار (Alphabetical Order) سے

کیا جاتا ہے۔ حروف تہجی کی ترتیب ASCII عدد پر مبنی ہے۔ مثال کے طور پر:

بیان	آؤٹ پٹ
"zero" < "one"	// false
"Zero" < "one"	// true
10 < 5	// false, numeric comparison.

جاوا اسکریپٹ کی مدد سے کلائنٹ سائڈ اسکریپٹنگ

```
"10" < "5" // true, string comparison.
"10" < 5 // false, numeric comparison;
"Ten" < 5 // Error occurs, "Ten" can not be
// converted into a number
```

10.7.4 منطقی آپریٹر

منطقی آپریٹر کا استعمال دو یا زیادہ شرائط کو یکجا کرنے کے لیے کیا جاتا ہے۔ جاوا اسکریپٹ میں مندرجہ ذیل تین منطقی آپریٹر ہوتے ہیں:

آپریٹر	تفصیل مثال کے ساتھ
&& (AND)	اگر دونوں آپرینڈ true ہیں تو true ظاہر کرتا ہے ورنہ false ظاہر کرتا ہے۔
(OR)	اگر دونوں آپرینڈ false ہیں تو false ظاہر کرتا ہے ورنہ true ظاہر کرتا ہے۔
! (NOT)	اگر آپرینڈ false ہے تو true ظاہر کرتا ہے ورنہ false ظاہر کرتا ہے

10.7.5 سلسلہ بندی آپریٹر

+ آپریٹر دو اسٹرنگ آپرینڈ کو ایک لڑی میں باندھ دیتا ہے۔ + آپریٹر اسٹرنگ آپرینڈ کو عددی آپرینڈ پر ترجیح (فوقیت) دیتا ہے۔ یہ بائیں سے دائیں کام کرتا ہے۔ نتیجہ اس ترتیب پر منحصر ہوتا ہے جس ترتیب میں عمل کو انجام دیا گیا ہے۔ مثال کے طور پر:

بیان	نتیجہ
"Good" + "Morning"	"GoodMorning"
"5" + "10"	"510"
"Lucky" + 7	"Lucky7"
4 + 7 + "Delhi"	"11Delhi"
"Mumbai" + 0 + 0 + 7	"Mumbai007"

10.7.6 مخصوص آپریٹر

مشروط آپریٹر (?:)

مشروط آپریٹر مخصوص جاوا اسکریپٹ آپریٹر ہے جس میں تین آپرینڈ ہوتے ہیں۔ اسی لیے اسے ٹرنری آپریٹر (Ternary Operator) بھی کہا جاتا ہے۔ مشروط آپریٹر شرط کے اعتبار سے متغیر کو قدر تفویض کر دیتا ہے۔

سنٹیکس

```
var_name = (condition) ? v_1 : v_2
```

اگر شرط درست ہے تو متغیر کو v_1 قدر تفویض کر دی جاتی ہے اور اگر شرط درست نہیں ہے تو متغیر کی قدر v_2

ہو جائے گی۔

مثال کے طور پر

```
status = (age >= 18) ? "adult" : "minor"
```

یہ بیان status متغیر کو "adult" قدر تفویض کر دیتا ہے اگر عمر 18 یا اس سے زیادہ ہے، نہیں تو یہ status متغیر کو "minor" قدر تفویض کر دیتا ہے۔

New

new آپریٹر کا استعمال واقعہ (Instance) کی تشکیل کرنے اور استعمال کنندہ کے ذریعہ متعین کیے گئے یا پہلے سے متعین آبجیکٹ ٹائپ کے لیے حافظہ مقرر کرنے کے لیے کیا جاتا ہے۔

سنٹیکس

```
ObjectName = new objectType ( param1 [,param2] ...[,paramN])
```

مثال

```
d = new Date(); // date assigns to object d
r = new rectangle(4, 5, 7, 8);
```

Delete

delete آپریٹر کا استعمال آبجیکٹ، آبجیکٹ پر اپریٹی array کے عنصر کو حذف کر کے **new** آپریٹر کے ذریعہ مقرر کردہ کی گئی میموری اسپیس کو خارج (De-allocate) کے لیے کیا جاتا ہے۔

سنٹیکس

```
delete object_name
delete object_name.property
delete array_name[index]
```

delete آپریٹر کا استعمال ایسے متغیرات کو مٹانے (Delete) کے لیے کیا جاتا ہے جنہیں ضمنی طور پر بیان کیا گیا ہو، تاکہ وہ جن کا اعلان **var** اسٹیٹمنٹ کے تحت کیا گیا ہو۔ اگر آپریشن ممکن ہے تو **delete** آپریٹر **true** ظاہر کر دیتا ہے اور اگر آپریشن ممکن نہیں ہے تو یہ **false** ظاہر کر دیتا ہے۔

```
a=12
var j= 63
myobj=new Number()
myobj.h=55 // create property h
delete x /* returns true (x is declared implicitly,
without using var) */
delete y /* returns false ( y is declared explicitly
using var) */
delete Math.PI /* returns false (cannot delete predefined
properties) */
delete myobj.h /* returns true (can delete user-defined
properties) */
delete myobj /* returns true (can delete if declared
implicitly) */
```

جاوا اسکرپٹ کی مدد سے کلائنٹ سائڈ اسکرپٹنگ

جب ہم کسی array element کو مٹاتے ہیں تو array کی لمبائی پر کوئی اثر نہیں ہوتا۔ مثال کے طور پر اگر ہم a[3] کو مٹاتے ہیں تو a[4] بدستور قائم رہتا ہے اور a[3] مٹ جائے گا۔ جب delete آپریٹر کسی array element کو مٹا دیتا ہے تو وہ ایلیمینٹ array سے خارج ہو جاتا ہے۔

this

جاوا اسکرپٹ میں this آپریٹر کا بھی استعمال ہوتا ہے۔ this آپریٹر موجودہ آبجیکٹ کو ظاہر کرتا ہے۔ یہ موجودہ آبجیکٹ کے لیے پوائنٹر کی طرح ہے۔

سنٹیکس

```
this[.propertyName]
```

مثال

اس آپریٹر کا استعمال عمر کی درستی کی جانچ کرنے کے لیے کیجیے۔ یہاں ان پٹ کو متن باکس (Text Box) کے ذریعہ فراہم کیا گیا ہے۔

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
    function validate(obj, min_age, max_age)
    {
        if ((obj.value < min_age) || (obj.value > max_age))
            alert("Invalid age for the Job!!!");
    }
</script>
</HEAD>
<BODY>
<B>Enter the age (between 18 and 40):</B>
<INPUT TYPE = "text" NAME = "age" SIZE = 2
onChange="validate(this, 18, 40)">
</BODY>
</HTML>
```

اس مثال میں ہم نے onChange (event handler) کے ذریعہ validate() فنکشن کو متعین کیا ہے۔



شکل 10.9

10.7.7 پیش روی آپریٹر

آپریٹر کی تحسیب پیش روی (Precedence) کی پہلے سے متعین ترتیب کے لحاظ سے کی جاتی ہے۔ مندرجہ ذیل جدول آپریٹر کی سبقت (اعلیٰ تقدیم سے ادنیٰ کی ترتیب میں) کو ظاہر کرتی ہے:

جدول 10.1: آپریٹر پیش روی

آپریٹر	تفصیل	تقدیم (ترجیح)
*	ضرب	بیش ترین ↑ کم ترین
/	تقسیم	
%	ماڈیولس	
+	جمع	
-	گھٹا (تفریق)	
<	کم سے کم	
<=	کم یا مساوی	
>	زیادہ سے زیادہ	
>=	زیادہ یا مساوی	
==	مساویت	
!=	مساوی نہیں	
&&	منطقی AND	
	منطقی OR	
?:	مشروط	
=	تفویض آپریٹر	
+=		
-=		
*=		
/=		
%=		
,	کوما	

10.8 جاوا اسکریپٹ پوپ اپ باکس (ڈائیلاگ باکس)

جاوا اسکریپٹ میں ونڈو آبجیکٹ کے تین طریقوں (Methods) کا استعمال کر کے تین قسم کے پوپ اپ باکس - انتباہی خانہ (Alert box) توثیقی خانہ، (Confirm Box) اور ہدایتی خانہ (Prompt box) کی تشکیل کی جاسکتی ہے۔

10.8.1 انتباہی خانہ

ونڈو آ بیکٹ کا (Alert) میٹھڈ ایک چھوٹے ڈائیلاگ باکس کی تشکیل کرتا ہے جس میں ایک مختصر پیغام اور "OK" کمانڈ بٹن ہوتا ہے، اسے Alert box کہا جاتا ہے۔ اس باکس میں متنبہ (Warning) کرنے والا آئیکن موجود ہوتا ہے۔

سنٹیکس

```
[window].alert("Text to be displayed on the popup box");
```

لفظ window اختیاری ہے۔

مثال

```
window.alert("I am to alert you about ....");  
or  
alert("I am to alert you about ....");
```

نتیجہ



alert box کا استعمال اس وقت کیا جاتا ہے جب استعمال کنندہ کو کچھ معلومات فراہم کرنا مقصود ہو۔ جب الارٹ باکس ظاہر ہوتا ہے تو آگے بڑھنے کے لیے استعمال کنندہ کو "OK" بٹن پر کلک کرنا لازمی ہے۔

Confirm Box 10.8.2

Confirm Box کا استعمال اس وقت کیا جاتا ہے جب استعمال کنندہ کو کسی معلومات کی تصدیق اور توثیق کرنا مقصود ہو۔ استعمال کنندہ کو "OK" یا "Cancel" بٹن پر کلک کرنا ہوگا۔

سنٹیکس

```
[window].confirm("Text to be confirmed");
```

مثال

```
confirm("Do you want to quit now?");
```

نتیجہ



Confirm Box بولین قدر کو ظاہر کرتا ہے۔ اگر استعمال کنندہ "OK" پر کلک کرتا ہے تو اس کا نتیجہ true کی شکل میں اور اگر "cancel" پر کلک کرتا ہے تو false کی شکل میں برآمد ہوتا ہے۔

Prompt Box 10.8.3

prompt box استعمال کنندہ سے input حاصل کرنے کے لیے استعمال کیا جاتا ہے۔ ہم مقررہ متن سے پہلے سے متعین کر سکتے ہیں۔ استعمال کنندہ کے ذریعہ (prompt) سے داخل کی گئی

معلومات کو متغیرہ میں اسٹور کیا جاسکتا ہے۔

سنٹیکس

```
prompt("Message" [, "default value in the text field"]);
```

مثال

```
var name = prompt("What's your name?", "Your name please...");
```



نتیجہ

جب استعمال کنندہ "OK" پر کلک کرتا ہے تو prompt box استعمال کنندہ کے ذریعہ داخل کی گئی اسٹرنگ کو اسٹور کر لیتا ہے اور اگر "cancel" پر کلک کرتا ہے تو یہ باکس صفر قدر کو اسٹور کر لیتا ہے۔

10.9 بلاک اسٹیٹمنٹ

مجملے بریکٹوں کا استعمال کر کے دو یا دو سے زیادہ بیانات کو یکجا کر کے واحد بلاک اسٹیٹمنٹ بنایا جاسکتا ہے۔

سنٹیکس

```
{
  statement_1
  statement_2
  .
  .
  statement_n
}
e.g. If (z > y)
{ x=10;
  y=20; }
```

10.10 براؤنچنگ اور لوپنگ اسٹیٹمنٹ

جاوا اسکریپٹ کسی اسٹیٹمنٹ یا اسٹیٹمنٹ کے بلاک کے نفاذ کو دہرانے یا متبادل طریقوں کا انتخاب کرنے کی سہولت فراہم کرتی ہے۔ جاوا اسکریپٹ براؤنچنگ کے لیے کچھ مشروط بیانات کی بھی سہولت فراہم کرتی ہے۔ مشروط بیان ایک ایسا بیان ہے جس کا استعمال شرط کے اعتبار سے کوڈ کے کسی حصہ کو نافذ کرنے یا اگر شرط کی تکمیل نہ ہو تو کسی دوسرے کام کو انجام دینے کے لیے کیا جاتا ہے۔ لوپنگ (Looping) بیانات کے سیٹ کا تکراری نفاذ ہے۔

10.10.1 10.10.1 براؤزنگ (مشروط) اسٹیٹمنٹ

If اسٹیٹمنٹ کے ساتھ براؤزنگ

If اسٹیٹمنٹ کا استعمال منطقی عبارت (شرط) پر مبنی کسی بیان یا بیانات کے بلاک پر عمل درآمد کرنے کے لیے کیا جاتا ہے۔ اس کی تین مختلف شکلیں ہیں:

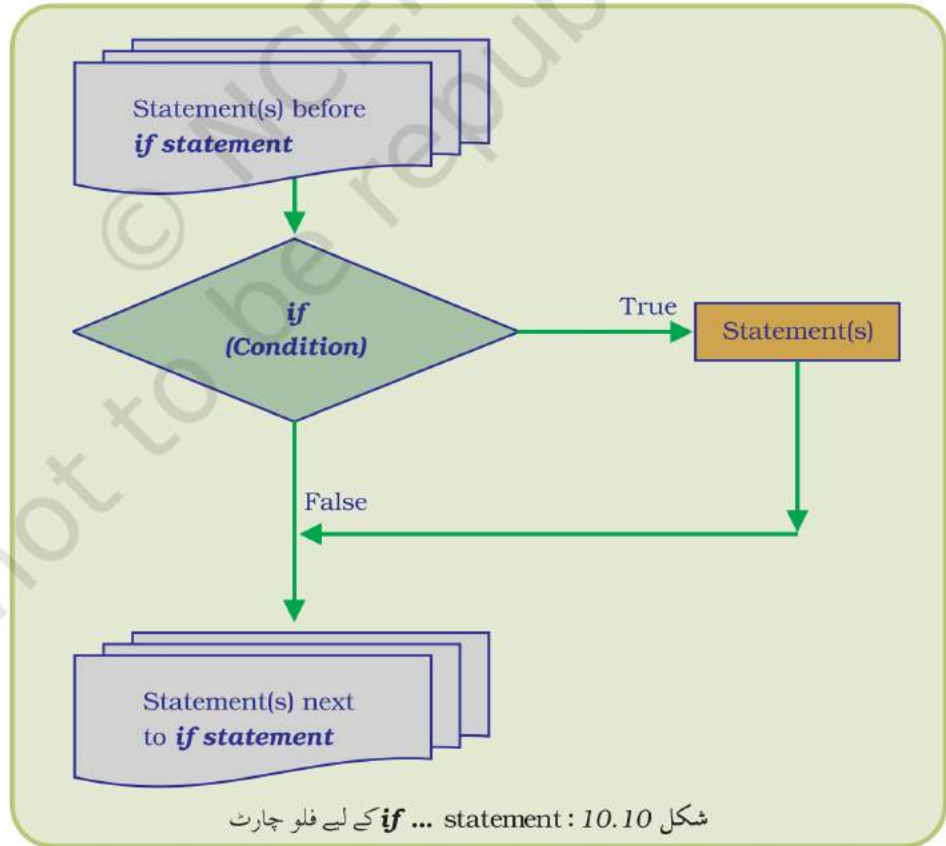
- if ..statement (simple **if** statement).
- if ..else statement.
- If .. else if .. else statement (**else if** ladder statement)

if ... statement

if ایک سادہ ترین فیصلہ کن بیان ہے۔ اس اسٹیٹمنٹ کا استعمال کسی بیان (یا بیانات) پر عمل درآمد کرنے کے لیے صرف اسی وقت کیا جاتا ہے جب متعینہ شرط صادق (True) آجائے۔

سینٹیکس

```
if ( condition )
{
  .. statement(s) to be executed if (condition) is true...
}
```

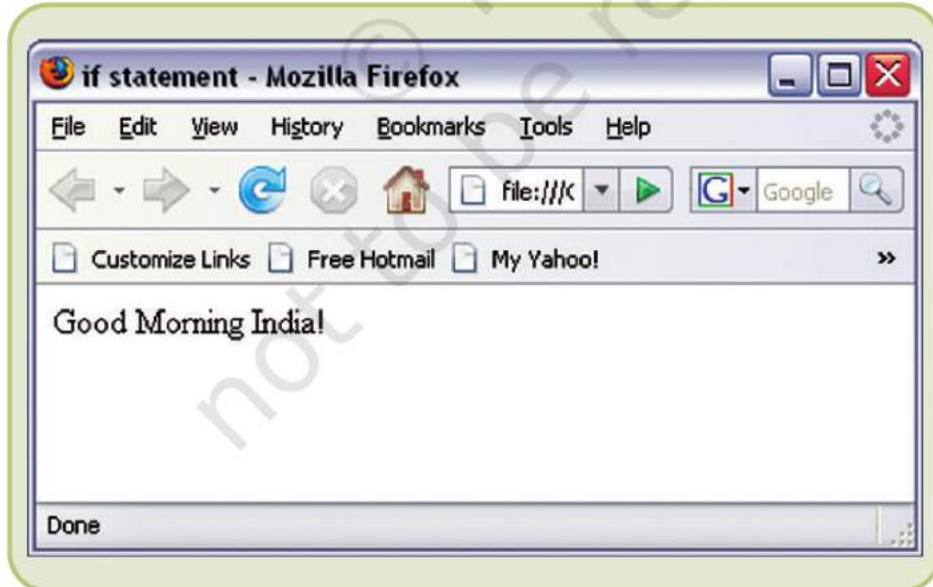


مذکورہ بالا فلو چارٹ میں، بیان (یا بیانات) پر عمل درآمد صرف اسی صورت میں ہوتا ہے جب منطقی بیان درست (True) ثابت ہوتا ہے، نہیں تو 'if statement' کے بعد والے بیان پر عمل کیا جائے گا۔
مثال: ایک جاوا اسکریپٹ پروگرام جو 'Good Morning India' کو ویب صفحہ پر صرف اسی وقت ظاہر کرتا ہے جب وقت 12 گھنٹے سے کم ہے نہیں تو صفحہ خالی رہے گا۔

```
<HTML>
<HEAD>
  <TITLE>if statement</TITLE>
</HEAD>
<BODY>
  <script type = "text/javascript">
    var d = new Date();
    var time = d.getHours(); // time stores hours
    if (time < 12)
    {
      document.write("Good Morning India!");
    }
  </script>
</BODY>
</HTML>
```

یہ HTML دستاویز صرف اسی وقت "Good Morning India" کو ظاہر کرے گا جب آپ کے سسٹم کا وقت 12 Hrs سے کم ہوگا نہیں تو آپ ویب صفحہ کو خالی پائیں گے۔

نتیجہ



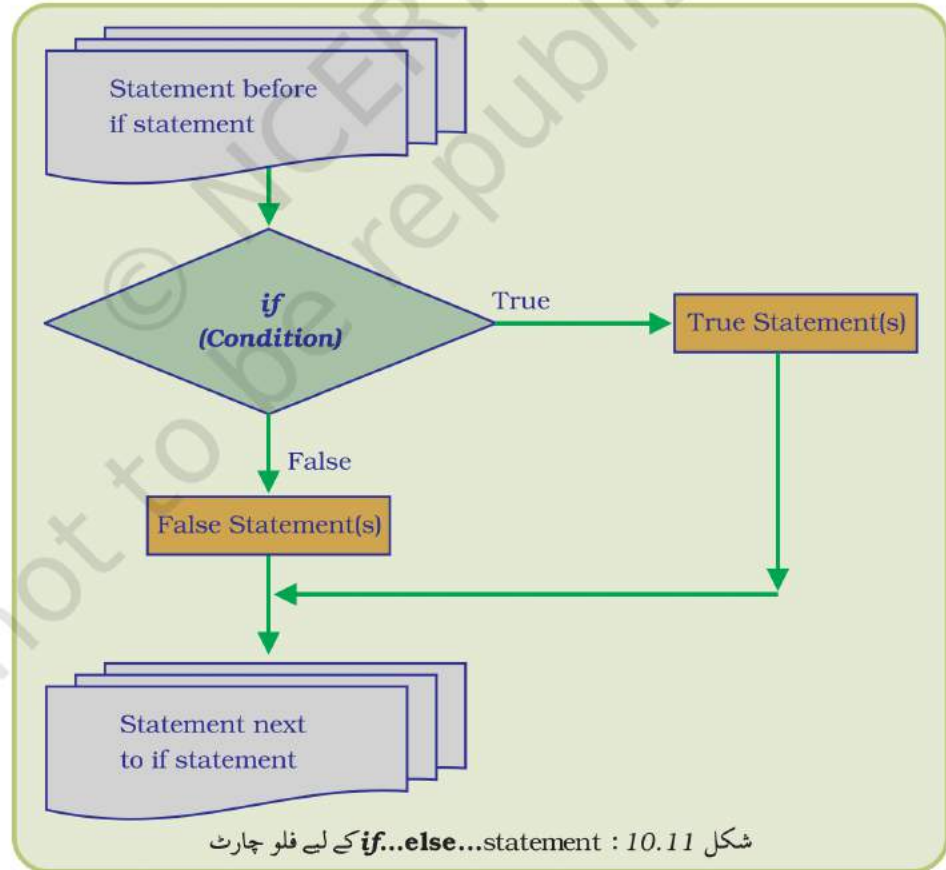
if...else...statement

یہ بیان سادہ if statement کی توسیع ہے۔ یہ منطقی جانچ پر مبنی دو بیانات یا بیانات کے مجموعہ میں سے کسی ایک بیان پر عمل درآمد کرنے کی اجازت دیتا ہے۔

سنٹیکس

```
if ( condition )
{
    True statement(s)...
}
else
{
    False statement(s)...
}
```

اگر شرط (Logical Expression) کی تکمیل ہو جاتی ہے تو true statement پر عمل درآمد ہوتا ہے نہیں تو false statement(s) پر عمل درآمد کیا جائے گا۔

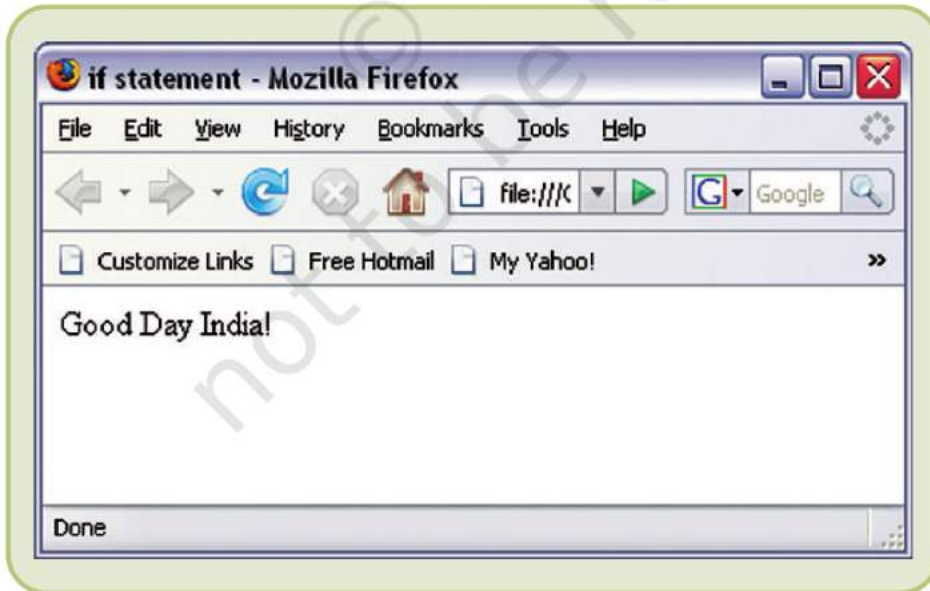


مثال: ایک جاوا اسکریپٹ پروگرام جو 'Good Morning India' کو ویب صفحہ پر صرف اسی وقت ظاہر کرتا ہے جب وقت 12 گھنٹے سے کم ہے نہیں تو "Good Day India" ظاہر ہو جائے گا۔

```
<HTML>
<HEAD>
  <TITLE>if else statement</TITLE>
</HEAD>
<BODY>
  <script type = "text/JavaScript">
    var d = new Date();
    var time = d.getHours();
    if (time < 12)
    {
      document.write("Good Morning India!");
    }
    else
    {
      document.write("Good Day India!");
    }
  </SCRIPT>
</BODY>
</HTML>
```

یہ HTML دستاویز صرف اسی وقت "Good Morning India" کو ظاہر کرے گا جب آپ کے سسٹم کا وقت 12 Hrs سے کم ہوگا۔ نہیں تو یہ "Good Day India" کو ظاہر کر دے گا۔

نتیجہ

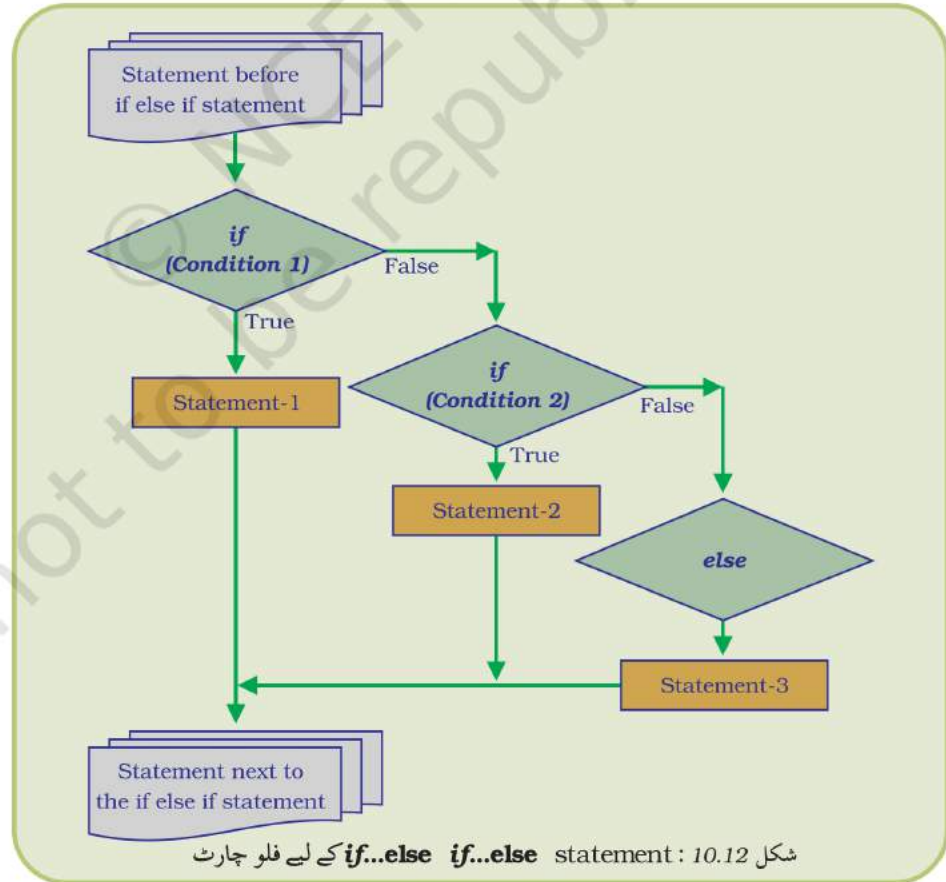


If...else if...else statement

if...else if...else بیان if...else if...else بیان کی مزید توسیع ہے۔ یہ بیان، بیانات کے مجموعہ میں سے کسی ایک بیان پر عمل درآمد کرانے کے لیے مفید ہے۔

سینٹیکس

```
if (condition1)
{
    code to be executed if condition1 is true
}
else if (condition2)
{
    code to be executed if condition2 is true
}
.
.
else
{
    code to be executed if any of the conditions is not true
}
```



مثال: ایک پروگرام لکھیے جس کے ذریعہ یہ جانچا جاسکے کہ prompt box میں داخل کیا گیا عدد صفر ہے، طاق ہے یا جفت ہے۔

```
<HTML>
<HEAD>
  <TITLE>Odd, Even or Zero</TITLE>
</HEAD>
<BODY>
<SCRIPT type = "text/JavaScript">
  var n = prompt("Enter your number:", "Type your number.
here");
  n = parseInt(n);          //converts string into number
  if (n == 0)
    alert("The number is zero!");
  else if (n%2)
    alert("The number is odd!");
  else
    alert("The number is even!");
</SCRIPT>
</BODY>
</HTML>
```

نتیجہ

سوئچ اسٹیٹمنٹ کے ذریعہ انتخاب

سوئچ اسٹیٹمنٹ (بیان) کا استعمال مختلف شرائط پر مبنی مختلف بیانات پر

عمل درآمد (Execute) کے لیے کیا جاتا ہے۔ یہ if...else

if...statement کے طویل سلسلہ کے مقابلے میں بہتر متبادل فراہم

کرتا ہے۔

سنٹیکس

```
switch (expression)
{
  case label1: //executes when value of exp. evaluates to label
    statements;
    break;
  case label2:
    statements;
    break;
  ...
  default : statements; //executes when non of the above labels
                      //matches the result of expression
}
```



شکل 10.13: آؤٹ پٹ

پروگرام 10.5 : جب آپ Prompt Box میں تاریخ داخل کرتے ہیں تو ہفتے کے دنوں کی تحسیب (لفظوں میں) کرنے کے لیے

```
<HTML>
<HEAD>
  <TITLE>Switch statement</TITLE>
</HEAD>
<BODY>
  <script type="text/JavaScript">
    // Enter date of birth to know the day of week on that day.
    var d=new Date(prompt("Enter your Date of Birth
    (e.g. November 17, 2002)", "Month DD, YYYY"))
    dy=d.getDay()
    switch (dy)
    {
      case 0:
        document.write("It was <b>Sunday</b> on that day.")
        break
      case 1:
        document.write("It was <b>Monday</b> on that day.")
        break
      case 2:
        document.write("It was <b>Tuesday</b> on that day.")
        break
      case 3:
        document.write("It was <b>Wednesday</b> on that day.")
        break
      case 4:
        document.write("It was <b>Thursday</b> on that day.")
        break
      case 5:
        document.write("It was <b>Friday</b> on that day.")
        break
      case 6:
        document.write("It was <b>Saturday</b> on that day.")
        break
      default:
        document.write("Please input a
        valid Date in?
        prescribed format !!!")
    }
  </script>
</BODY>
</HTML>
```



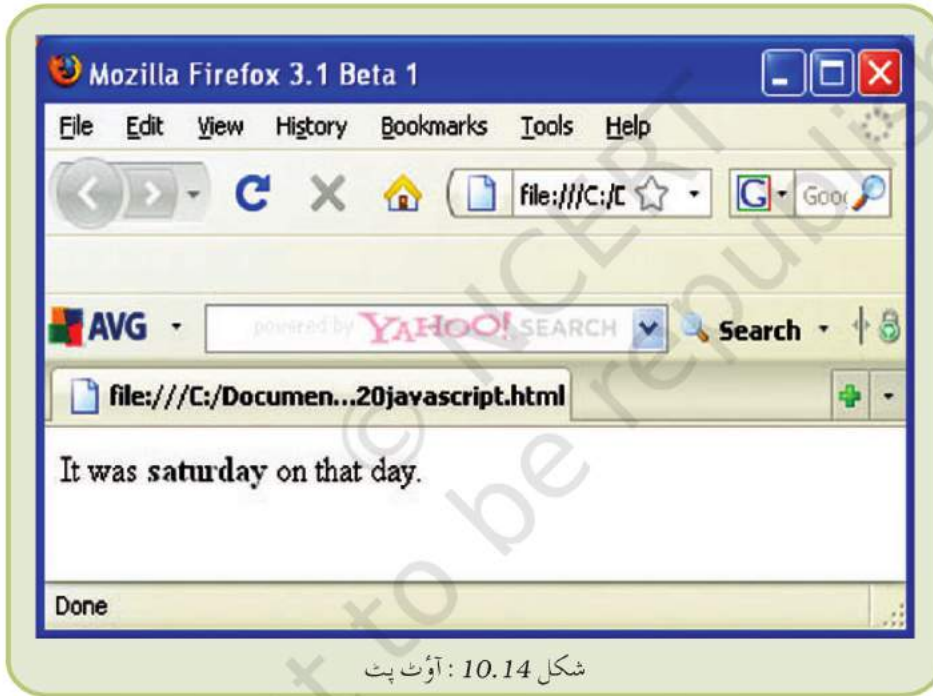
اس عبارت (Expression) کی قدر کا موازنہ سوئچ اسٹریکچر میں ہر کیس (Case) کی قدروں سے کیا جاتا ہے۔

اگر مطابقت ہو جاتی ہے تو اس کیس سے وابستہ کوڈ کے بلاک پر عمل درآمد ہوتا ہے۔ اگر کسی بھی کیس کی مطابقت نہیں ہوتی ہے تو ڈیفالٹ اسٹیٹمنٹ پر عمل درآمد ہوگا۔ اگلے کیس میں کوڈ کی خود کار طور پر انجام دہی کو روکنے کے لیے **break** کا استعمال کیجیے۔ ڈیفالٹ کیس کے اندر **break** کو استعمال کرنے کی ضرورت نہیں ہے۔

جب آپ "OK" بٹن کو دباتے ہیں تو ہمیں شکل 10.14 میں دکھایا گیا نتیجہ (Output) حاصل ہوگا۔

10.10.2 لوپ اسٹیٹمنٹ

لوپ اسٹیٹمنٹ ایک ایسا طریقہ کار (Mechanism) ہے جو جاوا اسکریپٹ انٹرپرائز کو کسی بیان (یا بیانات) پر بار بار اس وقت تک عمل درآمد کرنے کی ہدایت دیتا ہے جب تک کہ متعین شرط کی تکمیل نہیں ہو جاتی۔ جاوا اسکریپٹ میں مندرجہ ذیل لوپنگ اسٹیٹمنٹ کا استعمال ہوتا ہے:



شکل 10.14 : آؤٹ پٹ

- for
- do ... while
- while loop

زیادہ تر لوپ میں برعکس متغیرات ہوتے ہیں جو لوپ کے شروع ہونے سے پہلے کام کرنا شروع کر دیتے ہیں اور پھر لوپ کی ہر تکرار سے پہلے تحسیب شدہ شرط (عبارت) کے حصہ کے طور پر اس کی جانچ کی جاتی ہے۔ بالآخر برعکس متغیرہ میں ہر لوپ باڈی کے اخیر میں شرط کی دوبارہ تحسیب سے ٹھیک پہلے یا تو اضافہ ہوتا ہے یا اس کی تجدید ہو جاتی ہے۔

For

for لوپ تین اختیاری عبارتوں (Expressions) پر مشتمل ہوتا ہے جو ایک دوسرے سے سیکی کولن (:); کے ذریعہ علاحدہ رہتے ہیں اور ان کے بعد لوپ میں نافذ ہونے والے بیانات (Statements) کا بلاک ہوتا ہے۔ لوپ اسٹیٹمنٹ پر بار بار اس وقت تک عمل درآمد ہوتا رہتا ہے جب تک کہ شرط **false** ہے۔ **for** لوپ کا استعمال اس وقت ہوتا ہے جب ہمیں پہلے سے یہ معلوم ہو کہ اسکرپٹ کو کتنی مرتبہ چلایا جانا ہے۔

سنٹیکس

```
for([initial-expression]; [condition]; [increment-expression])
```

```
{
  statements
}
```

پیرامیٹر

- ابتدائی عبارت (Initial-expression) - اسے کاؤنٹر متغیر کو شروع کرنے کے لیے استعمال کیا جاتا ہے۔
- شرط (Condition) - اگر شرط (true) کی تکمیل ہو جاتی ہے تو اسٹیٹمنٹ پر عمل درآمد کیا جاتا ہے۔
- اضافی عبارت (Incr.-expression) - کاؤنٹر متغیر میں اضافہ (بڑھوتری) کرنے کے لیے۔

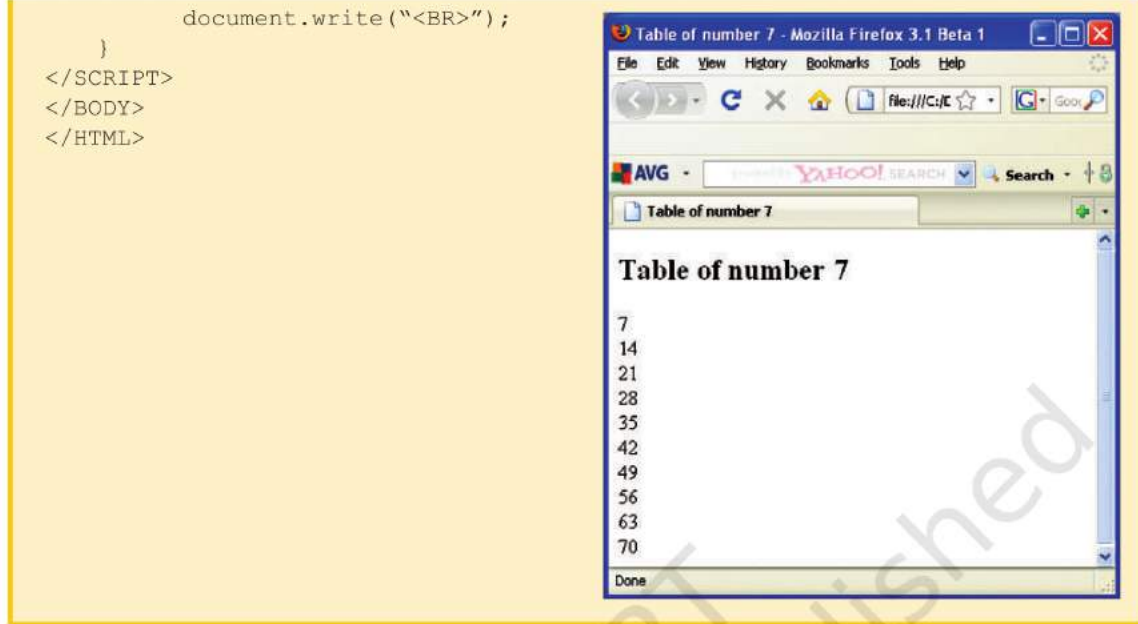
مثالیں

مندرجہ ذیل **for** اسٹیٹمنٹ متغیر **i** کو متعین کرتا ہے جس کی شروعات 1 سے ہوتی ہے۔ یہ اس بات کی جانچ کرتا ہے کہ 1 عدد 20 سے کم ہے، اگلے دو اسٹیٹمنٹ پر عمل درآمد کرتا ہے اور ہر مرتبہ لوپ سے ہو کر گزرنے کے بعد **i** میں 2 کا اضافہ کر دیتا ہے۔

```
// for loop to display odd numbers between 1 to 20
for (var i = 1; i < 20; i+=2)
{
  document.write(i);
  document.write("<BR>");
}
```

پروگرام 10.6: **for loop** اسٹیٹمنٹ کے ذریعہ 7 کا پہاڑہ تشکیل دینے کے لیے جاوا اسکرپٹ پروگرام۔

```
<HTML>
<HEAD>
  <TITLE> Table of 7 </title>
</HEAD>
<BODY>
<SCRIPT language="JavaScript" type="text/JavaScript">
  document.write("<H2> Table of number 7 </H2>");
  for(i = 1; i <= 10; i++)
  {
    document.write(7*i);
```



while

while لوپ اسٹیٹمنٹ **for** لوپ کے مقابلے میں زیادہ آسان ہے۔ یہ شرط اور اسٹیٹمنٹ کے بلاک پر مشتمل ہوتا ہے۔ لوپ سے ہو کر گزرنے سے پہلے ہر مرتبہ شرط کی تھیب کی جاتی ہے۔ اگر شرط کی تکمیل (True) ہو جاتی ہے تو یہ بلاک اسٹیٹمنٹ کو نافذ کر دیتا ہے۔

سنٹیکس

```
while (condition)
{
    statements
}
```

مثال: مندرجہ ذیل while لوپ سے بھی وہی نتیجہ حاصل ہوتا ہے جو گزشتہ مثال میں حاصل ہوا تھا۔

```
// While loop to display Odd numbers between 1 to 20
var i = 1; // Initialization of counter variable
while (i < 20) // Condition
{
    document.write(i);
    document.write("<BR>");
    i++; // Updation
}
```

while لوپ میں ہمیں ایک برعکس متغیرہ کو برقرار رکھنا پڑتا ہے جو لوپ میں اسٹیٹمنٹ پر عمل درآمد کو کنٹرول کرتا ہے۔

پروگرام 10.7: پہلے 10 فیبوناچی (Fibonacci) اعداد کی تشکیل کے لیے جاوا اسکریپٹ پروگرام لکھیے۔

```
<HTML>
<HEAD>
<TITLE>Fibonacci Series</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/JavaScript">
// Program to print the Fibonacci series upto 10 numbers
document.write("Fibonacci Series... upto 10 numbers <BR>".fontsize(4));
//.fontsize to increase the font size of the string
i=0;
document.write(i + " ");
j=1;
document.writeln(j + " ");
var x = 3;
while (x <= 10)
{
    t = i + j;
    document.write(t + " ");
    i = j;
    j = t;
    x++;
}
</SCRIPT>
</BODY>
</HTML>
```



Do...While

Do...While لوپ while لوپ کی ہی طرح ہے۔ یہ لوپ کو اس وقت تک دہراتا رہتا ہے جب تک کہ متعینہ شرط false نہیں ہو جاتی۔ لوپ ہمیشہ کم از کم ایک مرتبہ نافذ ہوگا حتیٰ کہ شرط false ہو، کیونکہ اسٹیٹمنٹ کے بلاک پر شرط کی جانچ ہونے سے پہلے عمل درآمد کیا جاتا ہے۔ اس لوپ میں اسٹیٹمنٹ اور منجھلے بریکٹ اختیاری ہیں۔

سینٹیکس

```
do
{
    statements
}
while (condition);
```

مثال: مندرجہ ذیل do...while لوپ سے بھی وہی نتیجہ حاصل ہوگا جو گزشتہ مثال میں برآمد ہوا تھا۔

```
// do...while loop to display Odd numbers between 1 to 20
var i = 1; // Initialization of counter variable
do
{
    document.write(i);
    document.write("<BR>");
    i++; // Updation
}
while (i < 20); // Condition
```

شرط کو اسٹیٹمنٹ کے بلاک کے بعد while کلیدی لفظ (Keyword) کے ساتھ قوسین میں لکھا جاتا ہے۔

10.10.3 لیبل (Label)

لیبل ایک شناخت کار (Identifier) ہے جس کے بعد کولن (:) لگا ہوتا ہے۔ یہ پروگرام کے بہاؤ کی رہنمائی میں مدد کرتا ہے۔

سنٹیکس

label: statement

لیبل کی قدر کوئی بھی جاوا اسکرپٹ شناخت کار ہو سکتا ہے۔ آپ لیبل کے ذریعہ جس اسٹیٹمنٹ کی شناخت کرتے ہیں وہ کوئی بھی اسٹیٹمنٹ ہو سکتا ہے۔

مثال

اس مثال میں "whileloop" لیبل while لوپ کی شناخت کرتا ہے۔

```
x=1;
whileloop: // Label
while (x<=10)
{
    document.write(x);
    x++;
}
```

10.10.4 بریک (Break)

Break اسٹیٹمنٹ کا استعمال سب سے اندر والے لوپ، سوچ اسٹیٹمنٹ یا لیبل کے ذریعہ نامزد کیے گئے اسٹیٹمنٹ سے باہر آنے کے لیے کیا جاتا ہے۔ یہ موجودہ لوپ کو ختم کر دیتا ہے اور اختیارات ختم کیے گئے لوپ کے بعد والے اسٹیٹمنٹ کو منتقل کر دیتا ہے۔

سنٹیکس

break[label]

بریک اسٹیٹمنٹ میں ایک اختیاری لیبل ہوتا ہے جو نامزد اسٹیٹمنٹ (Labeled Statement) سے باہر نکلنے کو کنٹرول کرتا ہے۔

مثال: مندرجہ ذیل پروگرام میں ایک بریک اسٹیٹمنٹ ہے جو while لوپ کو اس وقت ختم کر دیتا ہے جب یہ 3 کے مساوی ہو جاتا ہے۔

```
var i = 0;
while (i < 6)
{
    if (i == 3)
    break;           // the control moves out of loop in first iteration
    i++;
}
document.write(i);
```

10.10.5

continue اسٹیٹمنٹ اپنے بعد والے اسٹیٹمنٹ کو چھوڑ دیتا ہے یا نظر انداز (Skip) کر دیتا ہے اور اگلی تکرار والے لوپ پر عمل درآمد کرنے کی اجازت دیتا ہے۔ اسے while, do-while, for یا لیبل اسٹیٹمنٹ کے اندر if اسٹیٹمنٹ کے ساتھ استعمال کیا جاتا ہے۔

سنٹیکس

continue[label]

continue اسٹیٹمنٹ لوپ کو ختم نہیں کرتا ہے بلکہ while لوپ میں یہ شرط کی طرف واپس آ جاتا ہے اور for لوپ میں یہ update expression پر پہنچ جاتا ہے۔ continue اسٹیٹمنٹ میں ایک اختیاری لیبل شامل کیا جاسکتا ہے جو پروگرام کو کسی نامزد اسٹیٹمنٹ کو ختم کرنے اور متعینہ نامزد اسٹیٹمنٹ کو جاری رکھنے کی اجازت دیتا ہے۔

مثال

ایک پروگرام جس میں prompt() کا استعمال کر کے 50 ایلیمینٹ کو ان پٹ کیا گیا ہے اور continue اسٹیٹمنٹ کا استعمال کر کے 40 سے زیادہ والے نمبروں (Marks) کو جمع کیا گیا ہے۔

```
var marks = new Array();
var i = 0, sum=0;
while (i < 50)
{
    i++;
    // parseInt converts string value into a number.
    marks[i]=parseInt(prompt("Enter marks"));
    if (marks[i] <= 40) // when the condition is true then
    continue; // control goes to while condition expression.
    sum = sum + marks[i];
}
document.write(sum+"\n");
```

10.11 آجیکٹ کا ہندو بست کرنے والے اسٹیٹمنٹ

جاوا اسکرپٹ کچھ ایسے مخصوص اسٹیٹمنٹ فراہم کرتی ہے جو آجیکٹ کی دیکھ بھال کرتے ہیں۔ ایسے دو قسم کے اسٹیٹمنٹ ہیں

with اور for...in

For....In 10.11.1

for...in اسٹیٹمنٹ کسی آجیکٹ کی تمام پراپرٹیز کے لیے متعین متغیرہ کی تکرار کرتا ہے۔

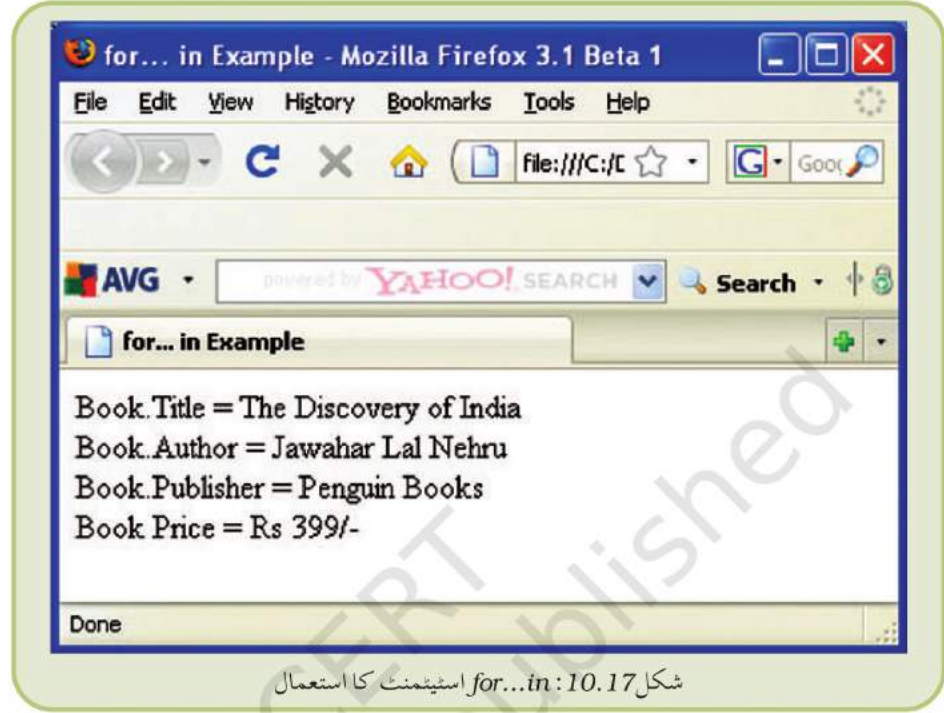
سنٹیکس

```
for (<variable>in <object>)
{
    statements
}
```

for...in اسٹیٹمنٹ کی باڈی پر آجیکٹ کی ہر ایک خاصیت کے لیے عمل درآمد کیا جاتا ہے۔ لوپ اسٹیٹمنٹ پر عمل درآمد سے پہلے آجیکٹ کی کسی ایک پراپرٹی کا نام متغیرہ کو اسٹرنگ کے طور پر تفویض کر دیا جاتا ہے۔ ہم اس متغیرہ کا استعمال [[آپریٹر کے ذریعہ آجیکٹ کی پراپرٹی کی قدر کو دیکھنے کے لیے کر سکتے ہیں۔

مثال: Book آجیکٹ کی ہر ایک خاصیت کی قدر کا نام چھاپنا۔

```
<HTML>
<HEAD>
<TITLE> for... in Example </TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
var Book = new Object(); // Object creation
// Properties and values of Book
Book = { Title:"The Discovery of India",
          Author:"Jawahar Lal Nehru",
          Publisher: "Penguin Books",
          Price:Rs 399/-
        };
var result = "";
// Name of distinct property of Book assign to b in each loop
execution
for (var b in Book)
{
    // Book[b] is used to get the values.
    result += "Book." + b + " = " + Book[b] + "<br>";
}
// To print names and values of each property of Book object.
document.write(result);
</SCRIPT>
</BODY>
</HTML>
```



شکل 10.17: for...in اسٹیٹمنٹ کا استعمال

With 10.11.2

جب ایک ہی آبجیکٹ کی خاصیتوں تک رسائی مقصود ہو تو with اسٹیٹمنٹ کا استعمال کر کے بہت زیادہ ٹائپنگ کرنے سے بچا جاسکتا ہے۔ مثال کے طور پر بہت زیادہ نیسٹڈ آبجیکٹ نظام مراتب (Nested object hierarchies) کے ساتھ کام کرنا ایک عام بات ہے۔ بعض اوقات ہمیں HTML فارم کے عناصر (Elements) تک رسائی کے لیے مندرجہ ذیل قسم کی عبارت (Expression) کو ٹائپ کرنا پڑتا ہے:

Frames[1].document.forms[0].fname.value

مثالیں

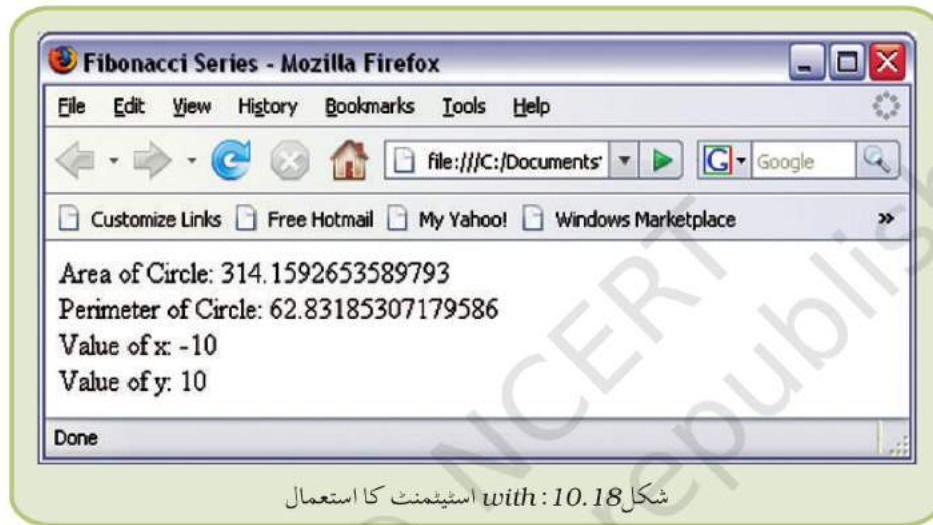
مندرجہ ذیل اسکرپٹ میں with اسٹیٹمنٹ کے استعمال کو ظاہر کیا گیا ہے۔ یہاں دو آبجیکٹ یعنی Math اور document فطری معروض ہیں۔ Math آبجیکٹ کے ساتھ with اسٹیٹمنٹ کے بعد والے اسٹیٹمنٹ PI خاصیت اور cos اور sin میٹھڈس سے منسوب ہیں۔ اسی طرح document آبجیکٹ کے ساتھ with اسٹیٹمنٹ کے بعد والے اسٹیٹمنٹ آبجیکٹ کے صراحت کے بغیر write میٹھڈ سے منسوب ہیں۔ جاوا اسکرپٹ ان حوالوں کے لیے Math اور document آبجیکٹ کو ٹیوٹ رکھتی ہے۔

```
var area, circumference
var r=10
with (Math)
```

کمپیوٹر اور مواصلاتی ٹیکنالوجی

```
{
    area = PI * r * r
    circumference = 2*PI*r
}
with (document)
{
    write("Area of the Circle: "+area+"<br>");
    write("Circumference of the Circle: "+circumference);
}
```

نتیجہ



شکل 10.18: with اسٹیٹمنٹ کا استعمال

10.12 جاوا اسکریپٹ فنکشن

فنکشن (Function) اسٹیٹمنٹ کا نامزد بلاک ہے جسے صرف اس کا نام لکھ کر بار بار نافذ کیا جاسکتا ہے اور یہ ایک قدر کو ظاہر کر سکتا ہے۔ یہ پروگرام کو ماڈیولر (Modular) اور قابل فہم بنانے کے لیے مفید ہے۔

10.12.1 فنکشن کا تعین کرنا

فنکشن کو مندرجہ ذیل سنٹیکس کا استعمال کر کے متعین کیا جاسکتا ہے:

سنٹیکس

```
function <function-name>([<parameter list>])
{
    ..body of the function ..
}
```

The function definition begins with keyword function, followed by function's name and optional parameter-list within parenthesis.. Braces

are used ({and }) to enclose all of the statement in a function. Let's an example of function definition.

```
function Welcome()
{
    alert("Welcome to NCERT ");
}
```

یہ اسٹینٹ *Welcome* فنکشن کو متعین کرتے ہیں جو استعمال کنندہ کے لیے *alert message* کو ظاہر کر دیتا ہے۔

10.12.2 فنکشن کے ساتھ پیرامیٹر استعمال کرنا

نظیری قدروں کی فہرست میں فنکشن کے ذریعہ حاصل کیے جانے والے آرگومنٹ (Arguments) پیرامیٹر کہلاتے ہیں۔ ان قدروں کو فنکشن کے اندر مقامی متغیروں (Local variables) کو تفویض کیا جاسکتا ہے۔ آئیے پیرامیٹر کا استعمال کرنے والے فنکشن کی ایک مثال پر غور کرتے ہیں۔

```
function Welcome(name)
{
    alert("Welcome to NCERT , " + name);
}
```

ہم نے یہ سیکھ لیا ہے کہ سادہ فنکشن کی تشکیل کس طرح کی جاتی ہے۔ HTML دستاویز میں <HEAD> سیکشن فنکشن کو متعین کرنے کی سب سے اچھی جگہ ہے کیونکہ اس سیکشن کے اسٹینٹ پر سب سے پہلے عمل درآمد ہوتا ہے اور اس طرح یہ یقینی بنایا جاسکتا ہے کہ فنکشن کو استعمال سے پہلے متعین کیا جائے۔

مثال: HTML دستاویز کے <HEAD> سیکشن میں فنکشن کو استعمال کرنا۔

```
<HTML>
<HEAD>
<TITLE> Define a Function </title>
<script language="JavaScript" type="text/JavaScript">
function Welcome(name)
{
    alert("Welcome to NCERT, " + name+"!");
}
</script>
</HEAD>
<BODY>
This is the body of a HTML document. You will observe that I have
used script code in small case and other HTML tags in Capital to
enhance the readability of JavaScript Code.
</BODY>
</HTML>
```

10.12.3 فنکشن کو طلب (Call) کرنا

فنکشن کو آرگیومنٹ کی فہرست لسٹ کے ساتھ فنکشن کا نام لکھ کر طلب (Call) کیا جاسکتا ہے۔ فنکشن کال کا استعمال ایونٹ ہینڈلر کوڈ (Event handler code) کے اندر بھی کیا جاسکتا ہے۔

سنٹیکس

<function name> ([<parameter list>])

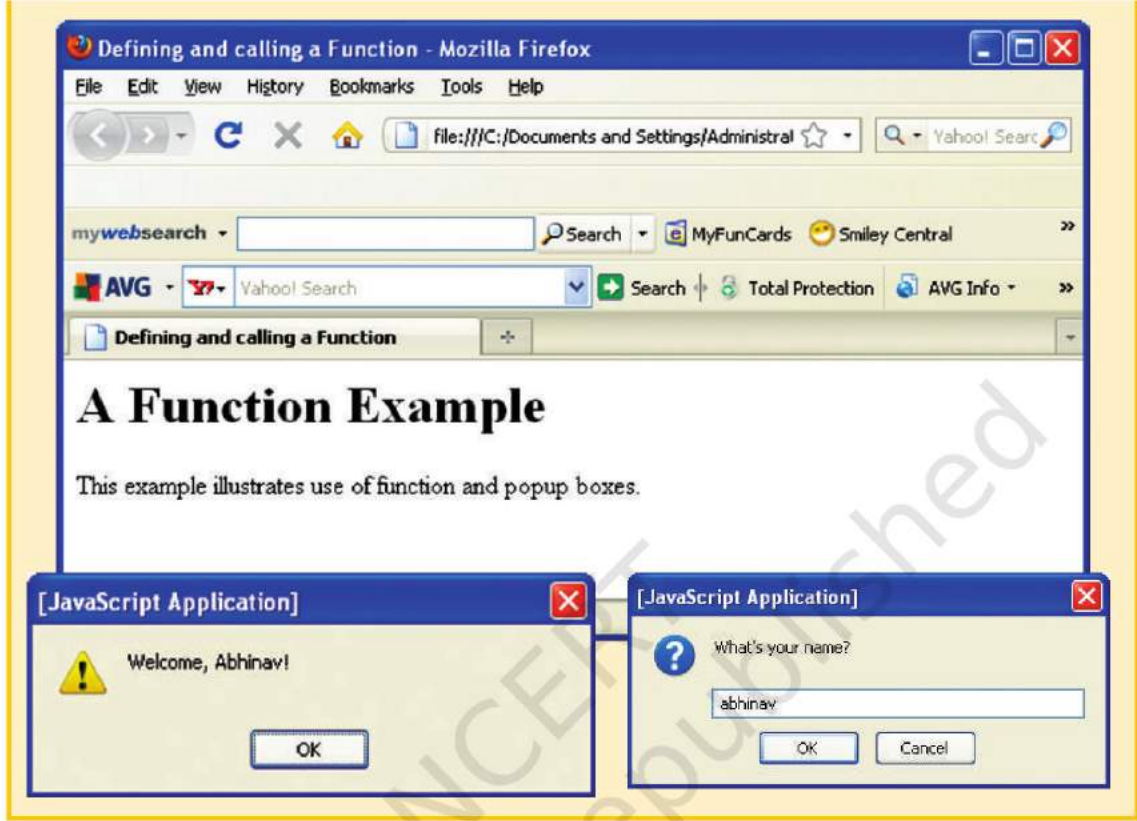
مثال

```
Welcome ("MANYA") ;
```

یہ جاوا اسکریپٹ انٹرپرائز کو فنکشن 'Welcome' کے پہلے اسٹینڈ کو اختیارات منتقل کرنے کی ہدایت دیتا ہے۔ یہ "MANYA" پیرامیٹر فنکشن کو بھی منتقل کرتا ہے۔ فنکشن کے اندر **name** متغیر کو قدر تفویض کر دی جائے گی۔

پروگرام 10.8: ایک HTML دستاویز جو فنکشن کے تعین اور Welcome() فنکشن کے طلب (Call) کرنے کو ظاہر کرتا ہے۔

```
<HTML>
<HEAD>
<TITLE> Defining and calling a Function </title>
<script language="JavaScript" type="text/JavaScript">
// function definition
function Welcome (name)
{
    alert ("Welcome, " + name);
}
</script>
</HEAD>
<BODY>
<H1> A Function Example</H1>
This example illustrates use of function and popup boxes.
<br>
<script language="JavaScript" type="text/JavaScript">
    var nm = prompt ("What's your name? ", "Your name please...");
    // Calling of function Welcome().
    Welcome (nm);
</script>
</BODY>
</HTML>
```



10.12.4 کسی وقوعہ (Event) سے فنکشن کو طلب (Call) کرنا

فنکشن کو متعین کرنے کے بعد اس کا استعمال وقوعات (مثلاً Click وقوعہ) کے ساتھ کیا جاسکتا ہے۔ مندرجہ ذیل میں فنکشن ایک اور جاوا اسکرپٹ کمانڈ بن جاتا ہے۔ مثال کے طور پر:

```
<INPUT type = "button"
value = "Calculate"
onClick = calc() >
```

جب استعمال کنندہ بٹن کو کلک کرتا ہے تو پروگرام خود بخود calc() فنکشن کو طلب کر لیتا ہے۔

10.12.5 فنکشن کی قدر کو ظاہر کرنا

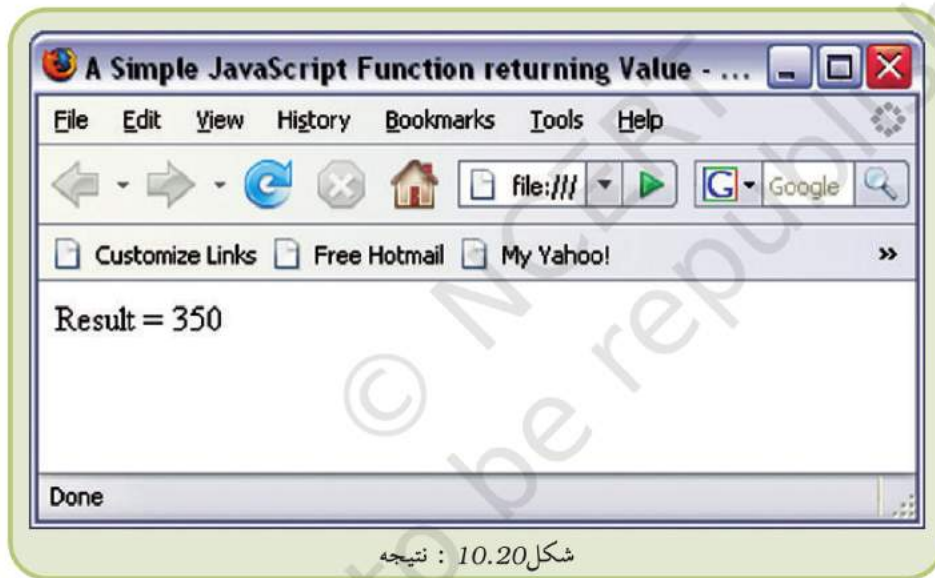
کسی فنکشن کی قدر کو ظاہر کرنے کے لیے return اسٹیٹمنٹ کا استعمال کیا جاتا ہے۔ یہ قدر اس اسٹیٹمنٹ آپریٹر کا استعمال کرنے والے متغیر میں آ جاتی ہے۔

مثال کے طور پر: فنکشن کا استعمال کر کے سادہ سود (Interest) کی تحسیب کرنے کے لیے پروگرام:

```
<HTML>
<HEAD>
<TITLE>A Simple JavaScript Function returning Value </TITLE>
```

```
<script language="JavaScript" type="text/JavaScript">
    function si( p, r, t )
    {
        var s = (p * r * t) / 100
        return s;    // function returning value s
    }
</script>
</HEAD>
<BODY>
<script language="JavaScript" type="text/JavaScript">
    var result = si( 1000, 5, 7)    // returned value assigned to
result
    document.write ( "Result = " + result);
</script>
</BODY>
</HTML>
```

نتیجہ



شکل 10.20 : نتیجہ

10.12.6 عیسید فنکشن

ایک فنکشن کو دوسرے فنکشن کے اندر متعین کیا جاسکتا ہے۔ یہ بھی ممکن ہے کہ انہیں اسٹیمٹ بلاک (مثلاً if اسٹیمٹ یا while لوپ وغیرہ کی باڈی) کے اندر متعین نہ کیا گیا ہو۔

مان لیجیے کہ ہم دائرہ کا رقبہ معلوم کرنا چاہتے ہیں جو $3.14 \times r^2$ کے مساوی ہے۔ عیسید فنکشن کو مندرجہ ذیل طریقہ سے لکھا

جائے گا:

```
function Area (r)
{
```

جاوا اسکرپٹ کی مدد سے کلائنٹ سائڈ اسکرپٹنگ

```
function Square(x)
{
    return x*x;
}
return 3.14*Square(r);
}
```

اس اسکرپٹ میں Square() فنکشن کو Area() فنکشن کے اندر متعین (نیسٹڈ) کیا گیا ہے۔

جاوا اسکرپٹ میں محفوظ / کلیدی الفاظ

abstract	boolean	break	byte
case	catch	char	class
const	continue	debugger	default
delete	do	double	else
enum	export	extends	false
final	finally	float	for
function	goto	if	implements
import	in	instanceof	int
interface	long	native	new
null	package	private	protected
public	return	short	static
super	switch	synchronized	this
throw	throws	transient	true
try	typeof	var	void
volatile	while	with	

خلاصہ

- جاوا اسکرپٹ پلیٹ فارم مشقی معروض پر مبنی اسکرپٹنگ لینگویج ہے۔
- ایسے جاوا اسکرپٹ کوڈ کو کلائنٹ سائڈ جاوا اسکرپٹ کا نام دیا گیا ہے جو کلائنٹ مشین پر ویب براؤزر کے ذریعہ نافذ کیے جاتے ہیں۔
- جاوا اسکرپٹ case-sensitive لینگویج ہے اور اس کے تمام بیانات (Statement) چھوٹے حروف میں لکھے جاتے ہیں۔
- جاوا اسکرپٹ میں جب بیانات کو علاحدہ سطروں میں رکھا جاتا ہے تو سی کولن (:) کو شامل نہیں کیا جاتا۔ اگر ہم بیانات کو ایک ہی سطر میں ملا دیتے ہیں تو انہیں علاحدہ کرنے کے لیے ہمیں سی کولن (:) کا استعمال کرنا ہوگا۔
- نتیجہ کوڈ ایکومنٹ ونڈو میں ظاہر کرنے کے لیے **document.write** ایک معیاری جاوا اسکرپٹ کمانڈ ہے۔
- حرفی علامات (Literals) وہ مستقل قدریں (Constant Values) ہیں جنہیں پروگرام کوڈ کے اندر براہ راست استعمال کیا جاتا ہے۔
- متغیرہ قدروں یا اسٹرنگ (String) پر مشتمل ہوتا ہے۔ متغیرہ میں اسٹوریج کی گئی قدروں کو متغیرہ کے نام کی مدد سے دیکھا جاسکتا ہے۔
- جاوا اسکرپٹ میں تین بنیادی ڈیٹا ٹائپ کا استعمال ہوتا ہے۔ یہ ڈیٹا ٹائپ ہیں: عدد (Number)، اسٹرنگ (String) اور بولین (Boolean)۔

- عبارت (Expression) آپریٹر اور حرفی علامتوں یا متغیرہ کے ناموں کی متحدہ شکل ہے۔
- اضافہ کرنے والے (Increment) یا تخفیف کرنے والے (Decrement) آپریٹر صرف ایک آپرینڈ (Operand) پر ہی عمل کرتے ہیں۔
- ان کا استعمال متغیرہ کی قدر میں 1 کا اضافہ کرنے یا 1 کم کرنے کے لیے کیا جاتا ہے۔
- اسائنمنٹ ایک ایسا بنیادی آپریٹر ہے جو اپنے دائیں آپرینڈ کی قدر بائیں آپرینڈ کو تفویض کر دیتا ہے۔ اس آپریٹر کو مساوی نشان (=) کے ذریعہ ظاہر کیا جاتا ہے۔
- نسبتی آپریٹر (Relational Operator) کا استعمال آپرینڈ کی قدروں کا موازنہ کرنے کے لیے کیا جاتا ہے اور یہ شرط (Condition) پر مبنی بولین قدر کو ظاہر کر دیتا ہے۔
- جاوا اسکریپٹ میں تین منطقی آپریٹر (Logical Operator) ہوتے ہیں: && (AND)، || (OR)، ! (NOT)۔
- دواسٹرنگ آپرینڈ کو ایک تسلسل میں پرونے کے لیے + آپریٹر کا بھی استعمال ہوتا ہے۔ یہ عددی آپرینڈ کے مقابلے میں اسٹرنگ آپرینڈ کو ترجیح دیتا ہے۔
- مشروط آپریٹر (Conditional Operator) (?:) ٹرنری (تہرا) آپریٹر بھی کہا جاتا ہے۔
- **new** آپریٹر کا استعمال استعمال کنندہ کے وضاحت کردہ آبجیکٹ ٹائپ یا پہلے سے متعین کیے گئے آبجیکٹ ٹائپ میں سے کسی ایک آبجیکٹ ٹائپ کے واقعہ (Instance) کی تشکیل کے لیے کیا جاتا ہے۔
- **delete** آپریٹر کا استعمال میموری اسپیس کو خارج کرنے کے لیے کیا جاتا ہے۔
- اگر متعین کی گئی خاصیت / انڈیکس متعین آبجیکٹ میں موجود ہے تو **in** آپریٹر **true** کو ظاہر کر دیتا ہے۔
- الارٹ باکس ایک ایسا ڈائیلاگ باکس ہے جو مقررہ پیغام (Text message) اور "OK" بٹن پر مشتمل ہوتا ہے۔
- **Confirm box** ایک ایسا باکس ہے جس کا استعمال کسی اطلاع کی تصدیق یا اسے قبول کرنے کے لیے کیا جاتا ہے۔ استعمال کنندہ کو آگے بڑھنے کے لیے "OK" یا "Cancel" بٹن پر کلک کرنا ہوتا ہے۔
- **Prompt box** کی مدد سے ہم استعمال کنندہ کے ذریعہ ٹیکسٹ فیلڈ کے ان پٹ کو حاصل کر سکتے ہیں۔
- بلاک اسٹیٹمنٹ دو یا دو سے زیادہ اسٹیٹمنٹ (بیان) کو ایک اسٹیٹمنٹ کی شکل میں یکجا کر دیتا ہے۔ بلاک اسٹیٹمنٹ کا استعمال عام طور سے مشروط (Conditional) اور لوپنگ (Looping) اسٹیٹمنٹ کے ساتھ کیا جاتا ہے۔
- **if** اسٹیٹمنٹ کا استعمال شرط کے مطابق کسی اسٹیٹمنٹ یا اسٹیٹمنٹ کے بلاک پر عمل درآمد کے لیے کیا جاتا ہے۔
- جاوا اسکریپٹ میں سوئچ اسٹیٹمنٹ کا استعمال مختلف شرائط کی بنیاد پر مختلف کاموں کو انجام دینے کے لیے کیا جاتا ہے۔ یہ کئی **if...else if...** اسٹیٹمنٹ کا بدل بھی ہو سکتا ہے۔
- لوپ اسٹیٹمنٹ کی مدد سے جاوا اسکریپٹ انٹر پریٹر ایک ہی اسٹیٹمنٹ کو اس وقت تک بار بار نافذ کرتا رہتا ہے جب تک کہ متعینہ شرط کی تکمیل نہیں ہو جاتی۔
- **for** لوپ تین اختیاری عبارتوں پر مشتمل ہوتا ہے۔ یہ بلاک اسٹیٹمنٹ کو اس وقت تک بار بار نافذ کرتا رہتا ہے جب تک کہ شرط **false** ہے۔
- **while loop** اسٹیٹمنٹ **for** لوپ کے مقابلے میں آسان ہے۔ یہ بلاک اسٹیٹمنٹ کو اس وقت تک دوہراتا رہتا ہے جب تک کہ شرط **false** ہے۔
- **while** لوپ کے برعکس **do...while** لوپ کوڈ کے بلاک کو ہمیشہ کم از کم ایک مرتبہ نافذ کرتا ہے۔

- لوپ سے باہر آنے کے بجائے، **continue** اسٹیٹمنٹ اگلے اسٹیٹمنٹ کو چھوڑ دیتا یا نظر انداز کر دیتا اور لوپ کی نئے سرے سے تکرار شروع کر دیتا ہے۔
- **for..in** اسٹیٹمنٹ متعینہ متغیرہ کو کسی آجیکٹ کی سبھی خاصیتوں کے لیے دوہراتا ہے۔
- **with** اسٹیٹمنٹ، ڈیفالٹ آجیکٹ کو بیانات کے ایک مجموعے (Set of statements) کے لیے مقرر (Establish) کرتا ہے۔
- فنکشن (Function) جاوا اسکرپٹ اسٹیٹمنٹ کے گروپ کی نامزد کاٹی ہے۔ اگر استعمال کنندہ فنکشن کو قدریں تفویض کرنا چاہتا ہے تو قدروں کو فنکشن کے نام کے بعد قوسین کے اندر رکھا جاتا ہے اور جب فنکشن کو طلب (Call) کیا جاتا ہے تو کوما کے ذریعہ علاحدہ آرگیومنٹ کی فہرست کے طور پر بھیج دیا جاتا ہے۔
- **return** اسٹیٹمنٹ کا استعمال فنکشن کی قدر کو ظاہر کرنے کے لیے کیا جاتا ہے۔ یہ قدر اس اسٹیٹمنٹ آپریٹر کا استعمال کرنے والے متغیرہ میں آ جاتی ہے۔
- فنکشن کو کسی دوسرے فنکشن کے اندر متعین (Nested) کیا جاسکتا ہے۔

مشقیں

مختصر جواب والے سوالات

- 1- کیا جاوا اسکرپٹ میں ایسی پبلیکیشن کے لیے پروگرام لکھنا ممکن ہے جو آزادانہ طور پر ایکزیکوٹ ہو سکتی ہے؟
 - 2- ہمیں اسٹیٹمنٹ میں سیسی کولن (;) کا استعمال کہاں کرنا چاہیے؟ کیا یہ لازمی ہے؟
 - 3- کلائنٹ سائڈ اور سرور سائڈ جاوا اسکرپٹ میں کیا فرق ہے؟
 - 4- جاوا اسکرپٹ کوڈ میں **document.write()** کا کیا مقصد ہے؟
 - 5- مندرجہ ذیل میں کون سے متغیرہ نام غیر موزوں (Invalid) ہیں؟ وجہ بتاتے ہوئے وضاحت کیجیے۔ **My_name**، **marks%**، **father's name**، **\$10**، **number10**
 - 6- محفوظ الفاظ (Reserved words) کیا ہیں؟ کیا ہم محفوظ الفاظ کا استعمال شناخت کار (Identifier) کے طور پر کر سکتے ہیں۔
 - 7- آپ **literals** سے کیا سمجھتے ہیں؟
 - 8- جاوا اسکرپٹ میں ڈیٹا ٹائپ کون کون سے ہیں؟
 - 9- **var** اسٹیٹمنٹ کا کیا مقصد ہے؟
 - 10- اسٹرنگ کو واحد یا دوہرے quotes میں کیوں رکھا جاتا ہے؟
 - 11- مندرجہ ذیل کو ظاہر کرنے والے جاوا اسکرپٹ کو دیکھیے:
- He said, "Mahatma Gandhi was a non-violent soldier of India."
- 12- کیا ہم مختلف ڈیٹا ٹائپ قدروں کو واحد **array** میں استعمال کر سکتے ہیں؟
 - 13- **stumarks** نام کے **array** میں 65, 75, 80, 87, 90 عددی قدروں کی تفویض کے لیے اسٹیٹمنٹ لکھیے۔
 - 14- کیا **null** اور صفر (zero) کی قدریں یکساں ہیں؟

15 - آپریٹر کیا ہوتے ہیں؟ آپریٹر کی کیا اہمیت ہے؟

16 - `prompt()` method کون سی قدر کو ظاہر کرتا ہے؟

17 - مندرجہ ذیل کے لیے معادل جاوا اسکریپٹ اسٹینٹ لکھیے:

$$\text{Area} = 3.14 r^2 \quad (\text{a})$$

$$\text{KE} = \frac{1}{2} mv^2 \quad (\text{b})$$

18 - متغیر `r` کی قدر کیا ہوگی؟

$$\text{var } r = 45 \% 7;$$

19 - متغیر `r` کی قدر کیا ہوگی؟

$$\text{var } a = 4;$$

$$\text{var } b = 7;$$

$$\text{var } c = \text{"NCERT"};$$

$$r = a + b + c;$$

20 - '==' آپریٹر اسٹینٹ آپریٹر '=' کے مساوی نہیں ہے (صحیح یا غلط)۔

21 - متغیر `total` کی قدر معلوم کیجیے۔

$$\text{total} = (360 * 5) + ((40 / 8) - 9) - ((14 * 6) / 2);$$

22 - متغیر `b` کی قدر کیا ہوگی؟

$$a = 25;$$

$$b = (++a) + 7;$$

طویل جواب والے سوالات

1 - ایک جاوا اسکریپٹ پروگرام لکھیے جس کے ذریعہ یہ معلوم کیا جاسکے کہ آیا دیا ہوا سال لوئڈ سال (Leap year) ہے یا نہیں۔ سال کے

ان پٹ کے لیے `prompt box` کا استعمال کیجیے۔

2 - ایک جاوا اسکریپٹ پروگرام لکھیے جو استعمال کنندہ کا مندرجہ ذیل کے ذریعہ استقبال کرتا ہے۔

Good Morning, <User_name> during hour in time is 4 to 11

Good Noon, <User_name> when hour value is 12.

Good Afternoon, <User_name> when hour value is 13 to 16

Good Evening, <User_name> when hour value is 17 to 23

Good Midnight, <User_name> when hour value is 24 or 1 to 3.

3 - دیے ہوئے تین اعداد میں سے سب سے بڑا عدد معلوم کرنے کے لیے ایک پروگرام لکھیے۔

4- جب مندرجہ ذیل جاوا اسکریپٹ کو نافذ کیا جائے گا تو کیا نتیجہ برآمد ہوگا؟

```
var get_res = confirm("Did you like this chapter?");
if (get_res == true)
alert("Okay! Let's go to the next!");
```

5- اگر **marks** متغیر 75 ہے تو مندرجہ ذیل کا نتیجہ کیا ہوگا؟

```
if (marks >= 80)
{
alert("Excellent!");
}
else if (marks >= 60 && marks < 80)
{
alert("Good!");
}
else if (marks > 50 && marks < 60)
{
alert("Average!");
}
else
{
alert("Improve yourself !");
}
```

6- سوئچ اسٹینٹ کا استعمال کرتے ہوئے 0 سے لے کر 9 تک کے اعداد کو الفاظ کے طور پر (مثلاً 3 کے لیے "Three") پرنٹ کرنے کے لیے ایک پروگرام لکھیے۔

7- ایک اسٹینٹ لکھیے جو ایک **alert box** ظاہر کرتا ہے، جیسا کہ ذیل میں دکھایا گیا ہے۔

8- **for** لوپ کا استعمال کر کے مندرجہ ذیل نتیجہ پرنٹ کرنے کے لیے ایک پروگرام لکھیے۔



1

22

333

4444

9- مندرجہ ذیل **for** لوپ کتنی مرتبہ نافذ ہوگا؟

```
for (a = 0; a <= 10; a = a + 2)
{
... statements ...
}
```

10- متغیر **sum** کی حتمی قدر کیا ہوگی؟

```
var sum = 0;
function add()
{
```

```
var sum = sum + 20;
}
add();
sum = sum + 1;
```

11۔ اسکرپٹ کے نافذ ہونے پر `alert box` میں کیا ظاہر ہوگا؟

```
var y = 0;
for (x = 0; x <= 5; x++, y = y + 50)
{
y = y + 10;
}
alert("The value of y is : " + y);
```

12۔ `prompt` کی مدد سے استعمال کنندہ کے ذریعہ داخل کیے گئے 15 اعداد کا اوسط معلوم کرنے کے لیے ایک پروگرام لکھیے۔

13۔ اگر `num` متغیر کی قدر 30 ہے تو مندرجہ ذیل `while` لوپ کتنی مرتبہ نافذ ہوگا؟

```
while (num <= 30)
{
... statements ...
num = num + 3;
}
```

14۔ کسی عدد کا الٹ (Reverse) (یعنی 123 کو 321) معلوم کرنے کے لیے ایک پروگرام لکھیے۔

15۔ کسی اعشاری عدد (Decimal) کو ثنوی عدد (Binary) میں تبدیل کرنے کے لیے ایک پروگرام لکھیے۔

16۔ یہ معلوم کرنے کے لیے کہ کوئی عدد مقلوب مستوی (Palindrome) ہے یا نہیں، ایک پروگرام لکھیے۔ کوئی عدد اپنے مقلوب عدد کے مساوی ہے تو وہ مقلوب مستوی کہلاتا ہے۔

17۔ `continue` کا استعمال کرتے ہوئے 1 سے 20 تک کے درمیان سبھی طاق اعداد کو ظاہر کیجیے۔

18۔ کسی عدد کے ہندسوں کا حاصل جمع معلوم کرنے کے لیے ایک پروگرام لکھیے۔ مثلاً 453 کا نتیجہ 4+5+3=12 ہوگا۔

19۔ ایک متعین حد تک مفرد اعداد (Prime Numbers) کی تشکیل کے لیے ایک پروگرام لکھیے۔

20۔ مندرجہ ذیل جاوا اسکریپٹ کوڈ کا نتیجہ کیا ہوگا؟

```
for(i=1; i<=5; i++)
{
document.write("<BR>")
for(j=1; j<=i; j++)
document.write("**")
}
```

21۔ مندرجہ ذیل کوڈ میں غلطی کی شناخت کیجیے:

```
function 3_alert_box
{
alert("Hi!, I am from a function.");
}
```

22۔ پہلے `n` اعداد کا حاصل جمع معلوم کرنے کے لیے `while` اسٹیٹمنٹ کا استعمال کرتے ہوئے ایک پروگرام لکھیے۔

جاوا اسکرپٹ کی مدد سے کلائنٹ سائڈ اسکرپٹنگ

23۔ مندرجہ ذیل کوڈ میں غلطی کی شناخت کیجیے:

```
fun_alert_box
{
document.write("Hi!, I am from a function");+1
}
```

24۔ نیسٹڈ فنکشن سے آپ کیا سمجھتے ہیں؟

متبادل جواب والے سوالات

1۔ بیرونی جاوا اسکرپٹ فائل کی توسیع ہے:

(i) .jscript

(ii) .js

(iii) .jav

(iv) .java

2۔ جاوا اسکرپٹ پروگرام میں تبصرے (Comments) کیوں استعمال کیے جاتے ہیں؟

(i) براؤزر کو یہ بتانے کے لیے کہ ہمارے جاوا اسکرپٹ کوڈ میں HTML ہے۔

(ii) کیونکہ HTML میں تبصروں کے لیے اپنی خود کی کمانڈ نہیں ہوتی۔

(iii) یہ واضح کرنے کے لیے کہ اسکرپٹ میں کیا ہے۔

(iv) ان میں سے کوئی نہیں

3۔ جاوا اسکرپٹ کا اصل نام تھا:

(i) JavaScript

(ii) LiveScript

(iii) WireScript

(iv) ECMAScript

4۔ مندرجہ ذیل میں سے کون سی عبارت (Expression) غیر موزوں ہے؟

(i) 7.5

(ii) b + a = c

(iii) {feet:10, inches:5}

(iv) [2,3,6,9,5,7]

5۔ مندرجہ ذیل عبارت (Expression) میں **res** کی قدر کیا ہوگی؟

var res = "70" + 25

(i) 95

7025 (ii)

25 (iii)

(iv) غلطی کی وجہ سے کوئی نتیجہ نہیں نکلتا گا

6- عبارت "bharat" < "India" کے ذریعہ کون سی قدر حاصل ہوگی؟

0 (i)

true (ii)

false (iii)

invalid expression (iv)

7- **delete** آپریٹر کا استعمال مندرجہ ذیل میں سے کسے مٹانے کے لیے کیا جاتا ہے؟

.js file (i)

cookies (ii)

created objects (iii)

function (iv)

8- مندرجہ ذیل میں سے کون سا آبجیکٹ **confirm() method** سے تعلق رکھتا ہے۔

(i) ڈاکیومنٹ (Document)

(ii) ونڈو (Window)

(iii) فریم (Frame)

(iv) تاریخ (Date)

9- مندرجہ ذیل کس اسٹیٹمنٹ کا استعمال اسٹیٹمنٹ کے بلاک کو بار بار نافذ کرنے کے لیے کیا جاسکتا ہے؟

(i) if...else...

(ii) Continue

(iii) While

(iv) Switch

10- اسٹیٹمنٹ نافذ ہونے کے بعد **sum** کی قدر کیا ہوگی؟

for(i=1, sum=0; i<=5; i++) sum+=i;

13 (i)

15 (ii)

25 (iii)

(iv) اسٹیٹمنٹ میں غلطی ہے۔

جاوا اسکرپٹ کی مدد سے کلائنٹ سائڈ اسکرپٹنگ

11 - مندرجہ ذیل میں سے کون سا لوپ کوڈ کے بلاک کو کم از کم ایک مرتبہ نافذ کرتا ہے؟

while (i)

for (ii)

do...while (iii)

for/in (iv)

12 - مندرجہ ذیل کس لوپ اسٹیٹمنٹ کا استعمال کسی آبجیکٹ کی تمام پراپریٹیز کے لیے متعینہ متغیر کی تکرار کے لیے کیا جاتا ہے؟

for (i)

do...while (ii)

for/in (iii)

ان میں سے کوئی نہیں (iv)

سرگرمیاں

- جاوا اسکرپٹ کوڈ کا استعمال کرتے ہوئے ایک HTML دستاویز لکھیے جو آپ کا نام، کلاس اور اسکول کا نام ظاہر کرتا ہو۔
- اپنے ویب صفحہ کے پس منظر کا رنگ تبدیل کرنے کے لیے جاوا اسکرپٹ کوڈ کا استعمال کرتے ہوئے ایک HTML دستاویز لکھیے۔ (اشارہ: document.bgcolor پراپٹی اور Math.random() فنکشن کا استعمال کیجیے۔)
- ایک پروگرام لکھیے جس سے اس بات کی جانچ کی جاسکے کہ استعمال کنندہ کو ووٹ دینے کا حق ہے یا نہیں۔ استعمال کنندہ ووٹ دینے کے لیے مندرجہ ذیل دونوں شرائط کو کی جانچ کرے گا:
 - (a) عمر 18 برس یا اس سے زیادہ ہونی چاہیے۔
 - (b) ہندوستان کا شہری ہونا چاہیے۔
- ایک سادہ کونز پروگرام لکھیے جو استعمال کنندہ سے 5 سوال پوچھتا ہے۔ سوالوں کے جواب کو **alert box** میں ظاہر کرتا ہے اور کونز کے اخیر میں استعمال کنندہ کو اسکور بتاتا ہے۔
- اپنی ویب سائٹ میں فارم (Form) کے درست ہونے کی جانچ کے لیے جاوا اسکرپٹ کوڈ کا استعمال کرتے ہوئے ایک HTML دستاویز لکھیے۔

حوالہ جات

JavaScript – The definitive Guide - David Flanagan, O'REILLY

SAMS Teach Yourself JavaScript in 24 hours - Michael Moncur,

Sams Publishing

JavaScript – A Beginner's Guide - John Pollock, McGraw-Hill.

<http://www.webreference.com/programming/javascript>

<http://www.webdevelopersnotes.com/tutorials/javascript>

کمپیوٹر اور مواصلاتی ٹیکنالوجی

<http://www.javascriptkit.com>

<http://www.w3schools.com/js>

<http://sun.com>

<http://www.mozilla.org>

© NCERT
not to be republished

© NCERT
not to be republished