12. JavaScript Switch Statement

The switch statement is used to perform different action based on different conditions.

12.1. The JavaScript Switch Statement

Use the switch statement to select one of many blocks of code to be executed.

Syntax

```
switch(n)
{
case 1:
  execute code block 1
  break;
case 2:
  execute code block 2
  break;
default:
  code to be executed if n is different from case 1
and 2
}
```

This is how it works: First we have a single expression *n* (most often a variable), that is evaluated once. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use **break** to prevent the code from running into the next case automatically.

Example

Display today's weekday-name. Note that Sunday=0, Monday=1, Tuesday=2, etc:

```
var day=new Date().getDay();
switch (day)
{
case 0:
  x="Today it's Sunday";
  break;
case 1:
  x="Today it's Monday";
  break;
```

```
    case 2:
      x="Today it's Tuesday";
      break;
    case 3:
      x="Today it's Wednesday";
      break;
    case 4:
      x="Today it's Thursday";
      break;
    case 5:
      x="Today it's Friday";
      break;
    case 6:
      x="Today it's Saturday";
      break;
    }
```

The result of *x* will be:

```
    Today it's Friday
```

12.2. The *default* Keyword

Use the *default* keyword to specify what to do if there is no match:

Example

If it is NOT Saturday or Sunday, then write a default message:

```
    var day=new Date().getDay();
    switch (day)
    {
    case 6:
      x="Today it's Saturday";
      break;
    case 0:
      x="Today it's Sunday";
      break;
    default:
      x="Looking forward to the Weekend";
    }
```

The result of *x* will be:

```
    Looking forward to the Weekend
```