HTML
CSS
Javascript

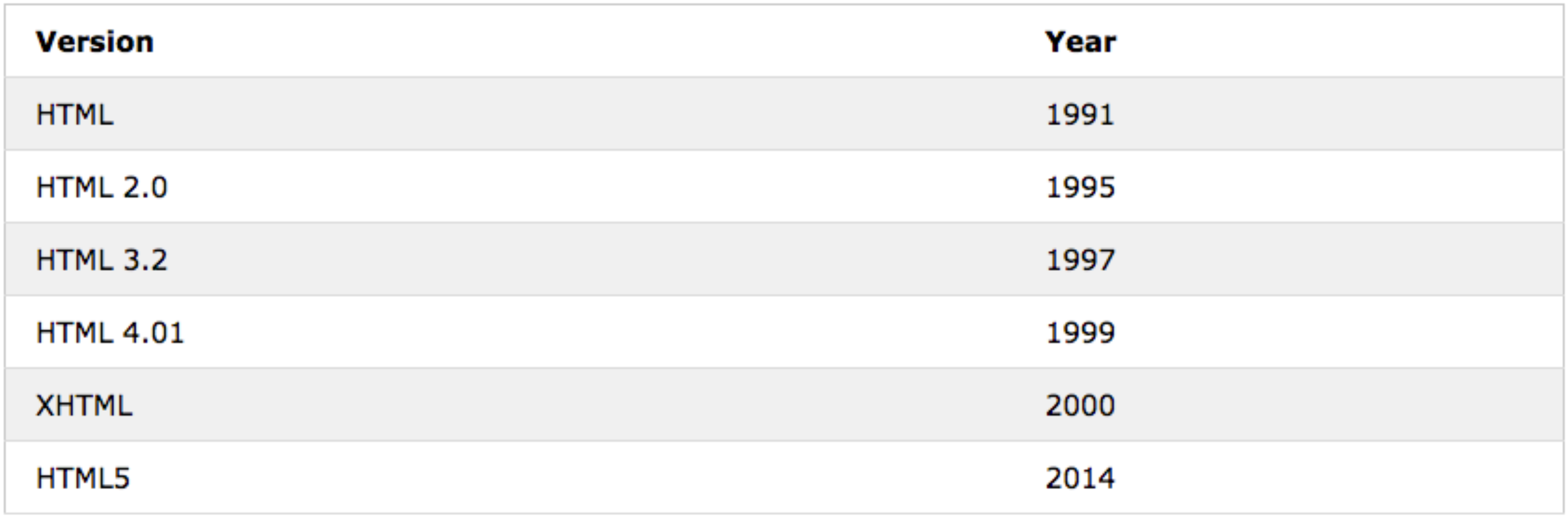
Bangsacerdas institute Batch 1 : 20 January 2017

Bangsacerdas institute

What is HTML ?

HTML (HyperText Markup Language) is not a programming language; it is a markup language, used to tell your browser how to structure the webpages you visit

HTML Version



Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

Anatomy of an HTML document

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My test page</title>
  </head>
  <body>
    <p>This is my page</p>
  </body>
</html>
```

HTML Head

the head's content is not displayed on the page. Instead, the head's job is to contain metadata about the document

ex :

```
<head>
  <meta charset="utf-8">
  <title>My test page</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="author" content="bangsacerdas">
  <meta name="description" content="bangsacerdas institute">
  <meta http-equiv="refresh" content="30">
  <meta property="og:title" content="Facebook Open Graph META Tags"/>
  <meta property="og:image" content="https://bangsacerdas/img/facebooklogo.png"/>
  <meta property="og:site_name" content="Bangsacerdas Blog"/>
  <meta property="og:description" content="online learning"/>
  <link rel="stylesheet" href="my-css-file.css">
  <script src="my-js-file.js"></script>
</head>
```

HTML ELEMENTS

An HTML element usually consists of a start tag and end tag

```
<tagname>Content goes here...</tagname>
```

HTML elements with no content are called empty elements. Empty elements do not have an end tag, such as the <br> element

The HTML5 standard does not require lowercase tags, but W3C recommends lowercase in HTML, and demands lowercase for stricter document types like XHTML.

Self-Closing Elements

```
<br>
<embed>
<hr>
<img>
<input>
<link>
<meta>
<param>
<source>
<wbr>
```

Nested HTML Elements

HTML elements can be nested (elements can contain elements)

```
<p>My cat is <strong>very</strong> grumpy.</p>
```

```
<p>My cat is <strong>very grumpy.</p></strong>
```

HTML Attributes

- ✓ All HTML elements can have **attributes**
- ✓ Attributes provide **additional information** about an element
- ✓ Attributes are always specified in the **start tag**
- ✓ Attributes usually come in name/value pairs like: **name="value"**

Ex :

```
<p title="I'm a tooltip">
This is a paragraph.
</p>
```

Core Attributes

- ✓ id

  ex :
  ```
  <p id="html">This para explains what is HTML</p>
  <p id="css">This para explains what is CSS</p>
  ```
- ✓ title

  ex :
  ```
  <h3 title="tooltips”>Titled Heading Tag Example</h3>
  ```
- ✓ class

  ex :
  ```
  <h3 class="class1 class2”>Titled Heading Tag Example</h3>
  ```
- ✓ style

  ex :
  ```
  <p style="font-family:arial; color:#FF0000;">Some text...</p>
  ```

Internationalization Attributes

✓ dir

ex :

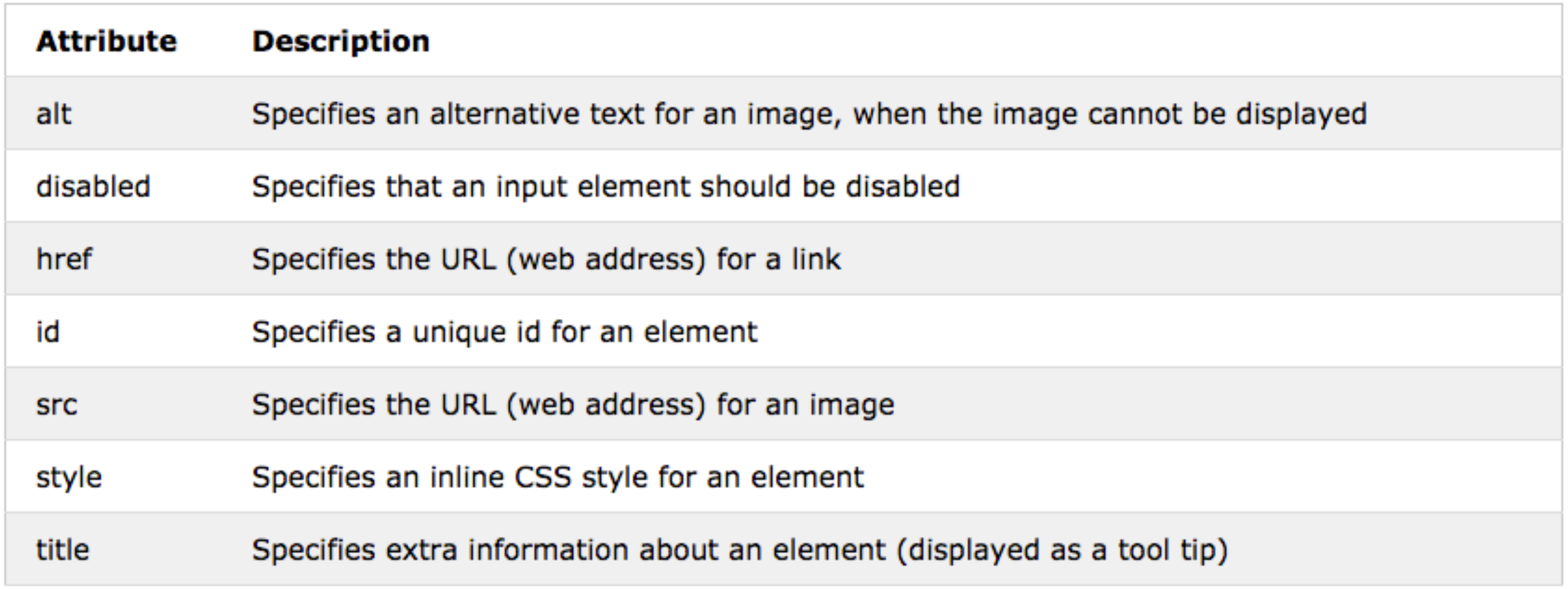
```
<!DOCTYPE html>
<html dir="rtl">
....
```

✓ lang

ex :

```
<!DOCTYPE html>
<html lang="en">
....
```

Common HTML Attributes



Attribute	Description
alt	Specifies an alternative text for an image, when the image cannot be displayed
disabled	Specifies that an input element should be disabled
href	Specifies the URL (web address) for a link
id	Specifies a unique id for an element
src	Specifies the URL (web address) for an image
style	Specifies an inline CSS style for an element
title	Specifies extra information about an element (displayed as a tool tip)

HTML Attribute Rules

- ✓ Use Lowercase Attributes
- ✓ Quote Attribute Values
- ✓ Single or Double Quotes?

HTML text fundamentals

BangsaCerdas.com

One of HTML's main jobs is to give text structure and meaning (also known as semantics,) so that a browser can display it correctly.

Basic HTML

Heading

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Paragraph

```
<p>Lorem ipsum dolor sit amet</p>
```

Basic HTML

List

Unordered

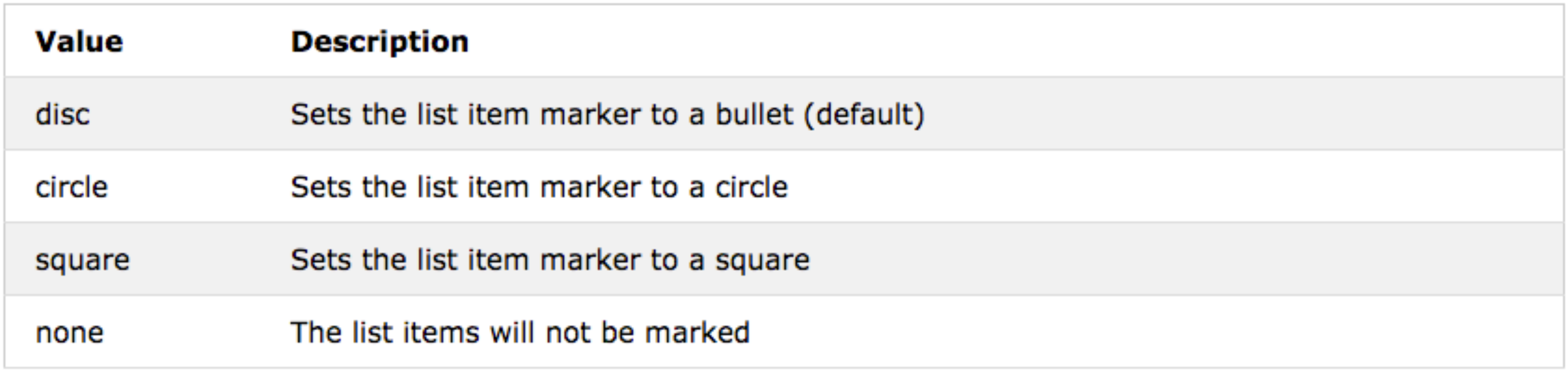
ex :

```
<ul>
    <li>milk</li>
    <li>eggs</li>
    <li>bread</li>
    <li>humous</li>
</ul>
```

Ordered

```
<ol>
    <li>milk</li>
    <li>eggs</li>
    <li>bread</li>
    <li>humous</li>
</ol>
```

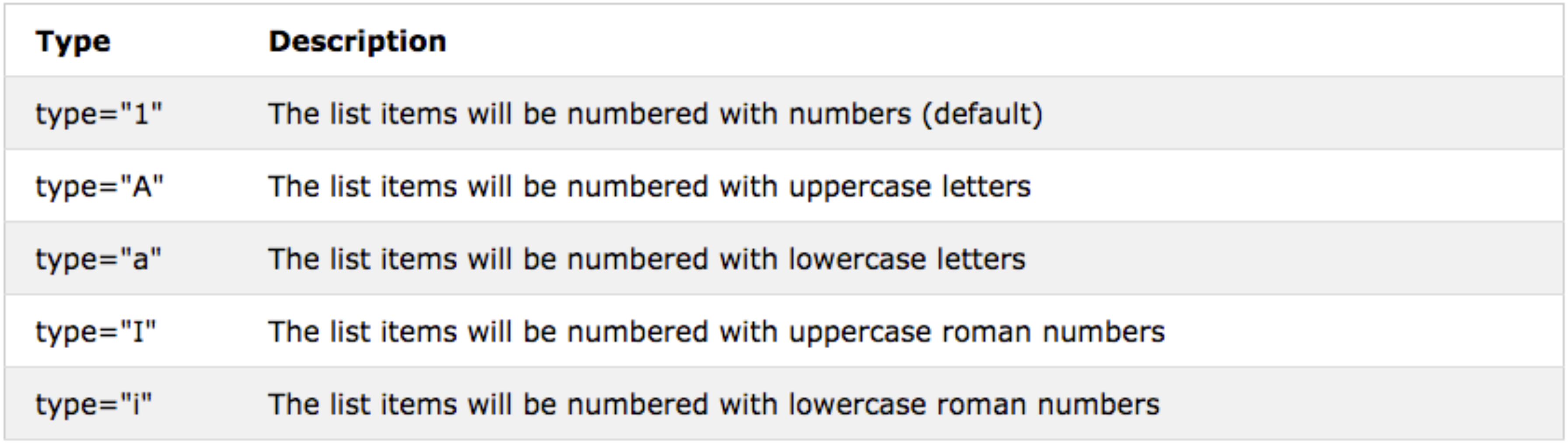
Basic HTML



Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Basic HTML

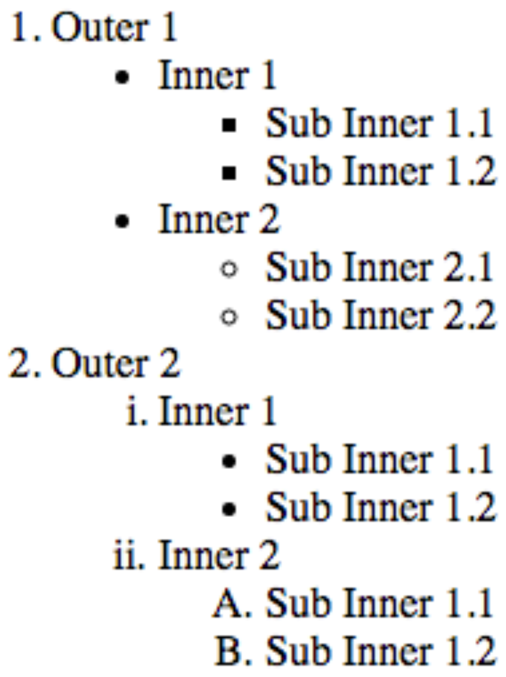
Ordered List type (Attributes)



Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Basic HTML

Study Case :



1. Outer 1
   - Inner 1
     - ▪ Sub Inner 1.1
     - ▪ Sub Inner 1.2
   - Inner 2
     - ◦ Sub Inner 2.1
     - ◦ Sub Inner 2.2
2. Outer 2
   - i. Inner 1
     - Sub Inner 1.1
     - Sub Inner 1.2
   - ii. Inner 2
     - A. Sub Inner 1.1
     - B. Sub Inner 1.2

Basic HTML

<b> - Bold text
<strong> - Important text
<i> - Italic text
<em> - Emphasized text
<mark> - Marked text
<small> - Small text
<del> - Deleted text
<ins> - Inserted text
<sub> - Subscript text
<sup> - Superscript text

HTML Comment

```
<!-- Write your comments here -->



<!--[if IE 9]>
    .... some HTML here ....
<![endif]-->
```

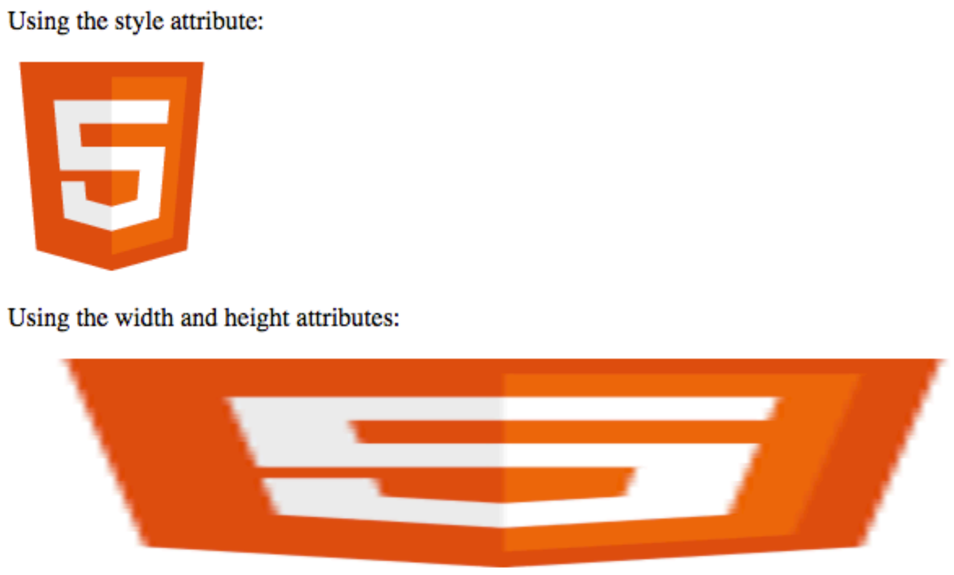
HTML Images

In HTML, images are defined with the `<img>`tag.

Syntax :

```
<img src="url" alt="some_text"
style="width:width;height:height;”>
```

HTML Images


Using the style attribute:

Using the width and height attributes:

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
    width:100%;
}
</style>
</head>
<body>

<p>Using the style attribute:</p>
<img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">

<p>Using the width and height attributes:</p>
<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

</body>
</html>
```

HTML Images

```
<p><strong>Float the image to the right:</strong></p>
<p>
<img src="smiley.gif" alt="Smiley face" style="float:right;width:
42px;height:42px;">
A paragraph with a floating image. A paragraph with a floating image.
A paragraph with a floating image.
</p>

<p><strong>Float the image to the left:</strong></p>
<p>
<img src="smiley.gif" alt="Smiley face" style="float:left;width:
42px;height:42px;">
A paragraph with a floating image. A paragraph with a floating image.
A paragraph with a floating image.
</p>
```

HTML Links

HTML links are hyperlinks.

You can click on a link and jump to another document

```
<a href="url">link text</a>
```

Default :

An unvisited link is underlined and blue

A visited link is underlined and purple

An active link is underlined and red

HTML Links

```
a:link {
    color: green;
    background-color: transparent;
    text-decoration: none;
}
a:visited {
    color: pink;
    background-color: transparent;
    text-decoration: none;
}
a:hover {
    color: red;
    background-color: transparent;
    text-decoration: underline;
}
a:active {
    color: yellow;
    background-color: transparent;
    text-decoration: underline;
}
```

HTML Links

The target attribute specifies where to open the linked document.

_blank - Opens the linked document in a new window or tab

_self - Opens the linked document in the same window/tab as it was clicked (this is default)

_parent - Opens the linked document in the parent frame

_top - Opens the linked document in the full body of the window

HTML Links

HTML bookmarks are used to allow readers to jump to specific parts of a Web page.

Bookmarks can be useful if your webpage is very long.

To make a bookmark, you must first create the bookmark, and then add a link to it.

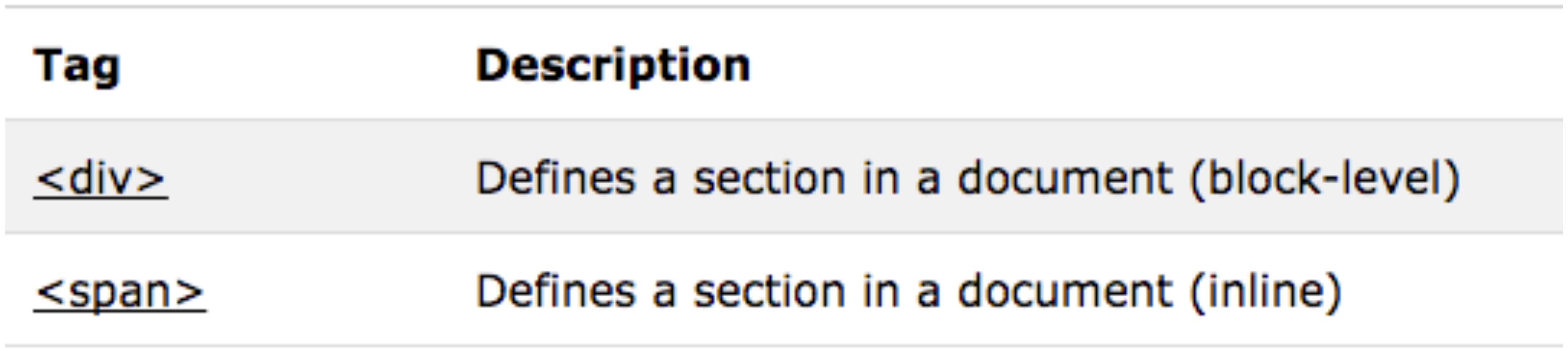
When the link is clicked, the page will scroll to the location with the bookmark.

HTML Tables

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the `<table>` tag in which the `<tr>` tag is used to create table rows and `<td>` tag is used to create data cells.

HTML Grouping tags



Tag	Description
<div>	Defines a section in a document (block-level)
<span>	Defines a section in a document (inline)

HTML CSS

CSS stands for Cascading Style Sheets.

CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

CSS saves a lot of work. It can control the layout of multiple web pages all at once.

CSS can be added to HTML elements in 3 ways:

- Inline - by using the style attribute in HTML elements
- Internal - by using a <style> element in the <head> section
- External - by using an external CSS file

HTML CSS

Inline

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

Internal

```
<style>
h1 {
    color: blue;
    font-family: verdana;
    font-size: 300%;

}
p  {
    color: red;
    font-family: courier;
    font-size: 160%;
}
</style>
```

External

```
<link rel="stylesheet" href="styles.css">
```

HTML Javascript

The <script> tag is used to define a client-side script (JavaScript).

The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

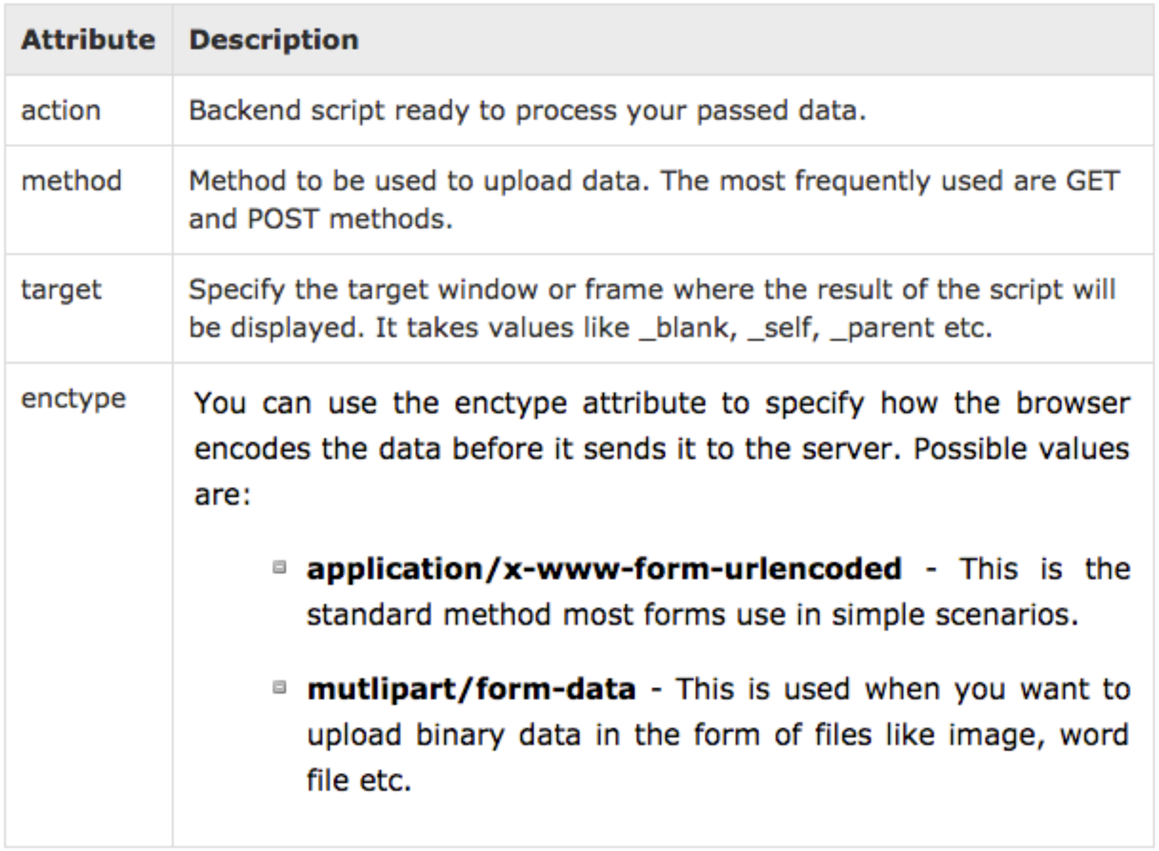
Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content

HTML Forms

HTML Forms are required when you want to collect some data from the site visitor. For example during user registration you would like to collect information such as name, email address, credit card, etc.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

HTML Form Attributes



Attribute	Description
action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
enctype	You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are: ▫ **application/x-www-form-urlencoded** - This is the standard method most forms use in simple scenarios. ▫ **mutlipart/form-data** - This is used when you want to upload binary data in the form of files like image, word file etc.

HTML Form Controls

Text Input Controls

Checkboxes Controls

Radio Box Controls

Select Box Controls

File Select boxes

Hidden Controls

Clickable Buttons

Submit and Reset Button

HTML5 Input Types

color
date
datetime
datetime-local
email
month
number
range
search
tel
time
url
week

HTML 5 Semantic

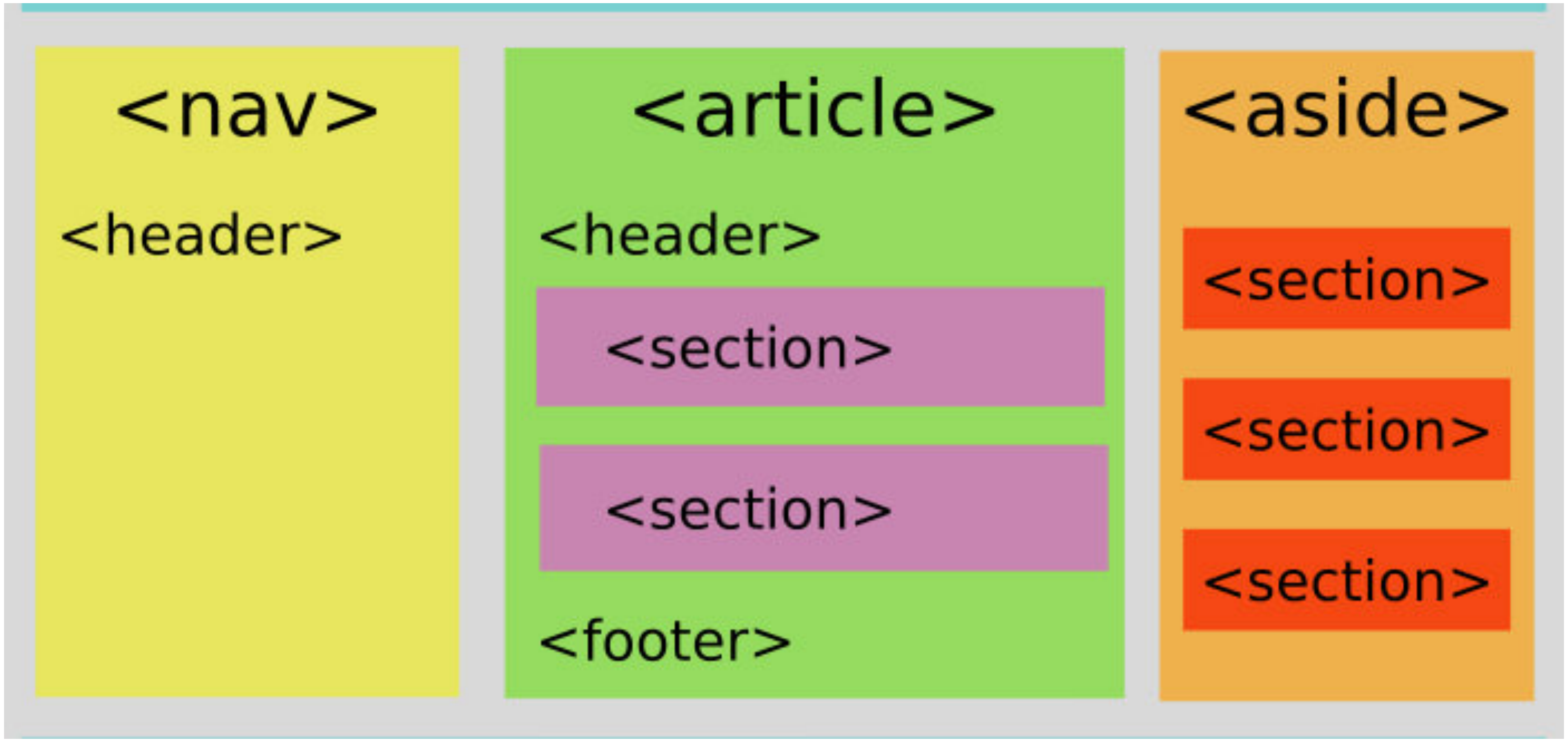
Semantics is the study of the meanings of words and phrases in a language

Semantic elements = elements with a meaning

A semantic element clearly describes its meaning to both the browser and the developer

Examples of non-semantic elements: `<div>` and `<span>` - Tells nothing about its content

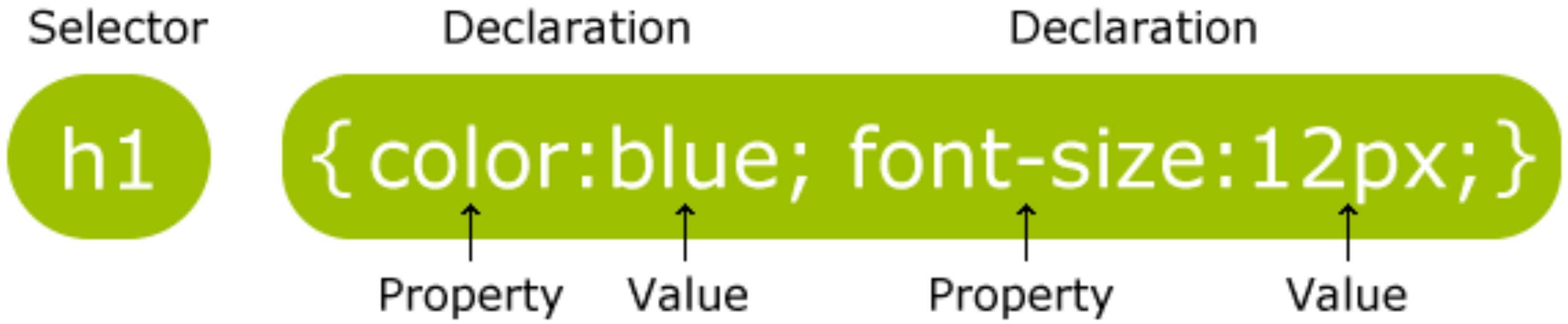
HTML 5 Semantic



What Is CSS

CSS, or Cascading Style Sheets, is a presentation language created to style the appearance of content

CSS Syntax


Selector
Declaration
Declaration
h1
{ color:blue; font-size:12px; }
Property
Value
Property
Value

CSS Selectors

The Universal selector

```
* {
   color: blue;
   background-color: white;
  }
```

The element Selector

```
p {
    text-align: center;
    color: red;
}
```

The id selector

```
 #para1 {
    text-align: center;
    color: red;
}
```

CSS Selectors

The class selector

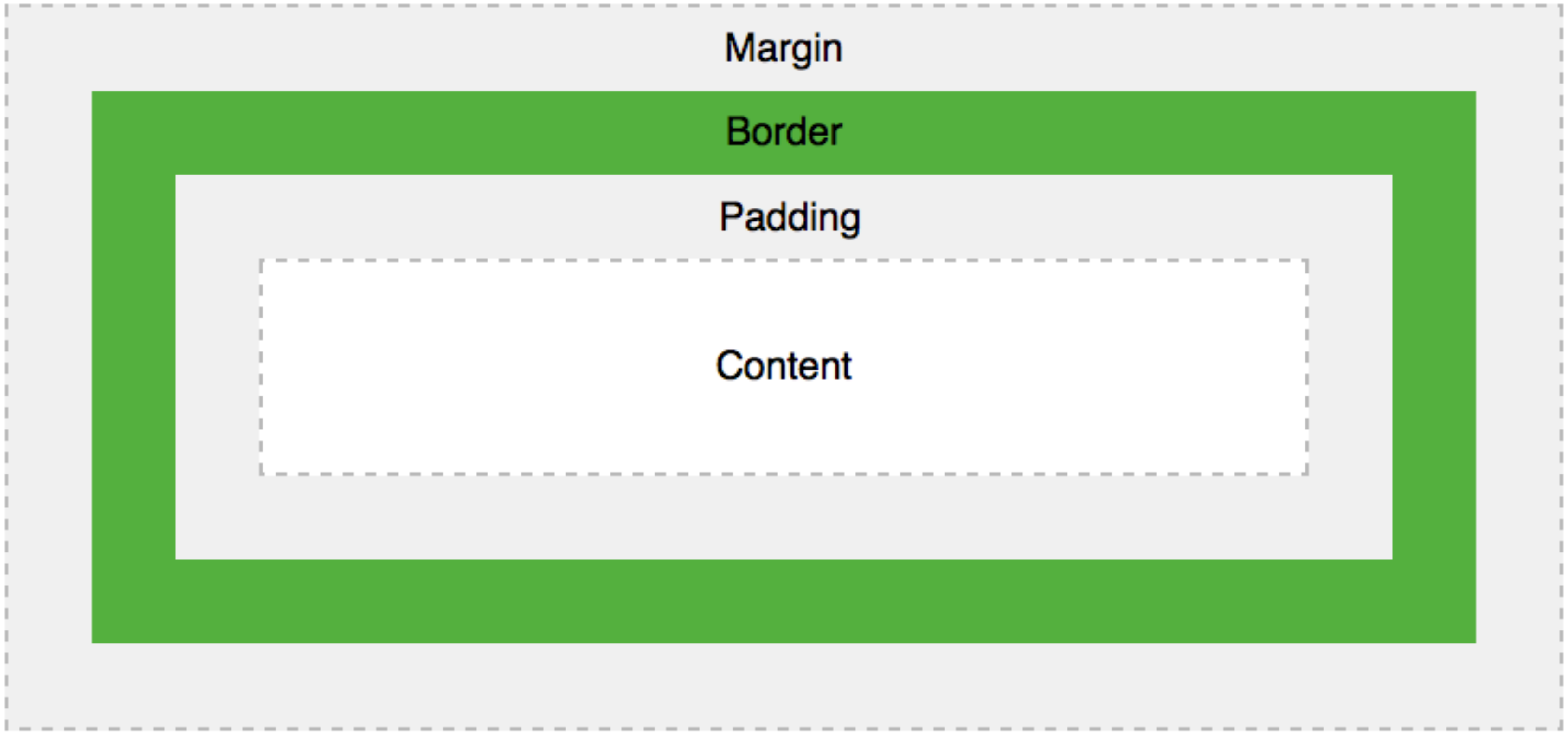
```
.center {
    text-align: center;
    color: red;
}

p.center {
    text-align: center;
    color: red;
}
```

The group selector

```
h1, h2, p {
    text-align: center;
    color: red;
}
```

CSS Box Model



CSS Shorthand

```
div {
    border: 1px solid black;
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
    background-color: lightblue;
}

div {
    border: 1px solid black;
    margin: 100px 150px 100px 80px;
    background-color: lightblue;
}
```

CSS Pseudo-class

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

Style an element when a user mouse over it
Style visited and unvisited links differently
Style an element when it gets focus

CSS Combinators

Descendant Selector

Child Selector

Adjacent Sibling Selector

General Sibling Selector

JavaScript

JavaScript is an object-based scripting language that is lightweight and cross-platform.

JavaScript is not compiled but translated. The JavaScript Translator (embedded in browser) is responsible to translate the JavaScript code.

3 Places to put JavaScript code

Between the body tag of html

Between the head tag of html

In .js file (external javaScript)

JavaScript Comment

```
// single comment
/*
Multiple comment
Multiple comment
Multiple comment
*/
```

JavaScript Variable

A JavaScript variable is simply a name of storage location. There are two types of variables in JavaScript : local variable and global variable

There are some rules while declaring a JavaScript variable (also known as identifiers).

- ✓ Name must start with a letter (a to z or A to Z), underscore(_), or dollar($) sign.
- ✓ After first letter we can use digits (0 to 9), for example value1.
- ✓ JavaScript variables are case sensitive, for example x and X are different variables.

Local VS Global

```
<script>
function abc(){
var x=10;//local variable
}
</script>
```

VS

```
<script>
var data=200;//gloabal variable
function a(){
document.writeln(data);
}
function b(){
document.writeln(data);
}
a();//calling JavaScript function
b();

</script>
```

JavaScript Data Types

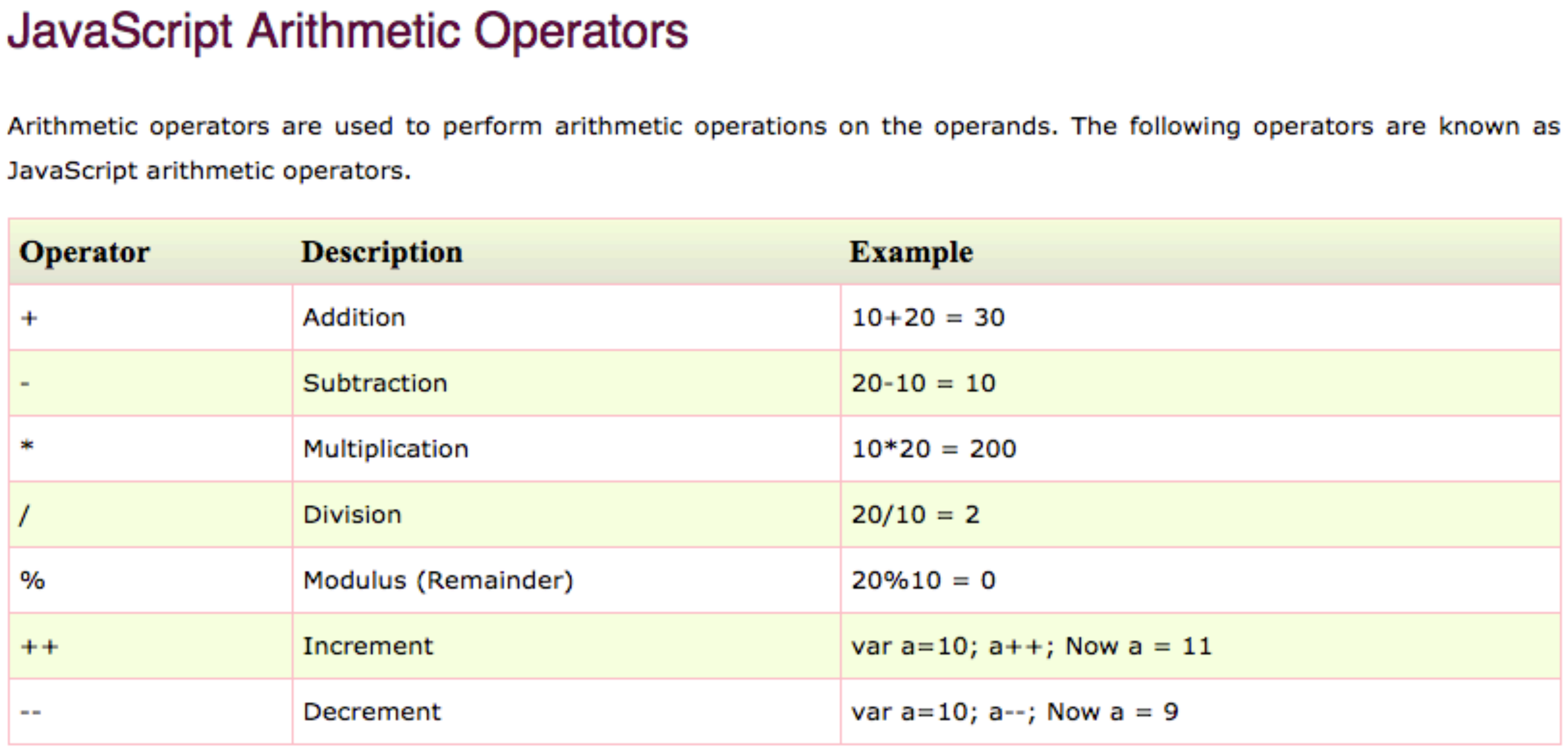
Javascript Data Types

JavaScript provides different data types to hold different types of values. There are two types of data types in JavaScript.

- ✓ Primitive data type
- ✓ Non-primitive (reference) data type

JavaScript Operators

Bangsa Cerdas.com

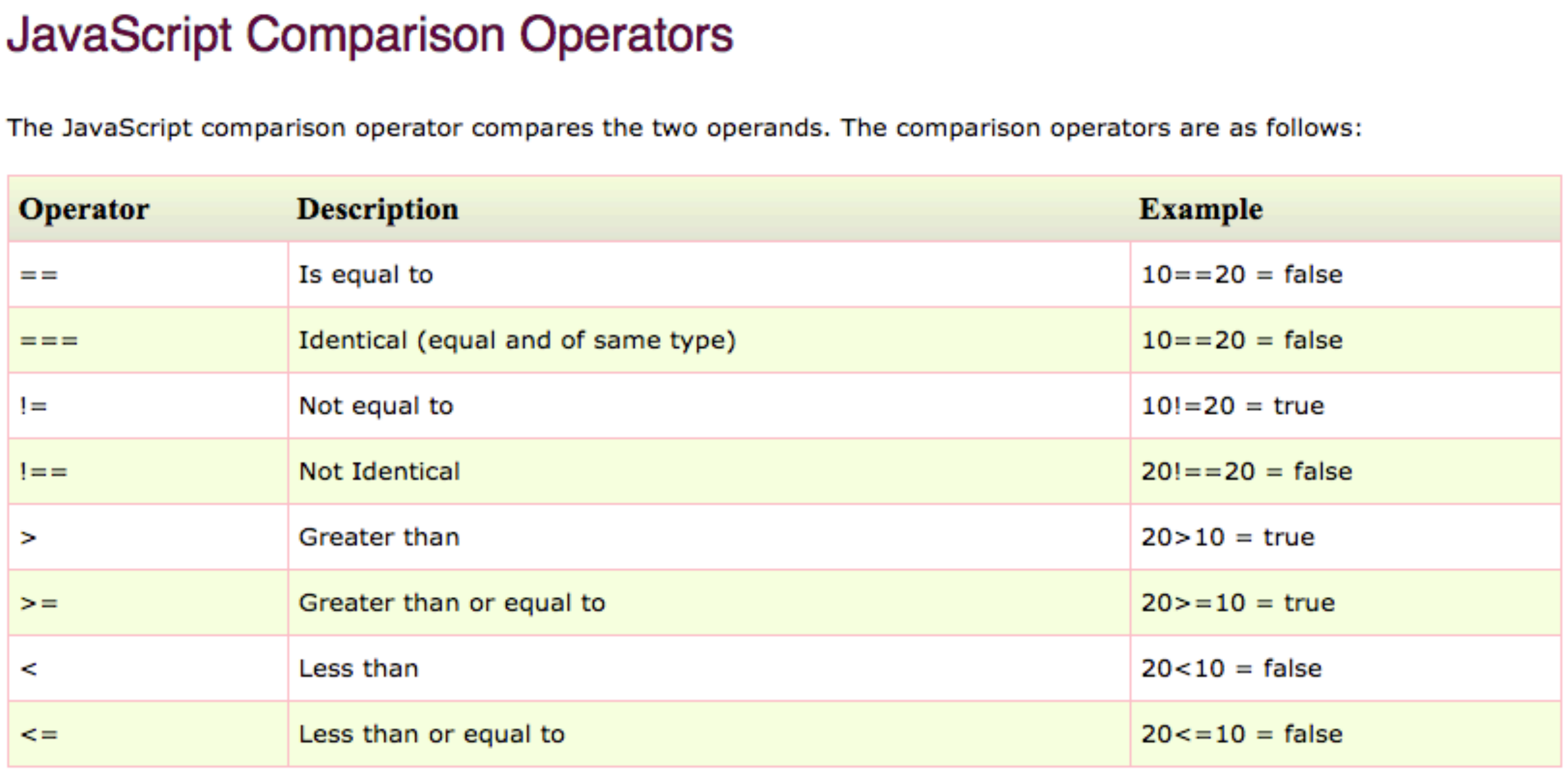


JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript arithmetic operators.

Operator	Description	Example
+	Addition	10+20 = 30
-	Subtraction	20-10 = 10
*	Multiplication	10*20 = 200
/	Division	20/10 = 2
%	Modulus (Remainder)	20%10 = 0
++	Increment	var a=10; a++; Now a = 11
--	Decrement	var a=10; a--; Now a = 9

JavaScript Operators

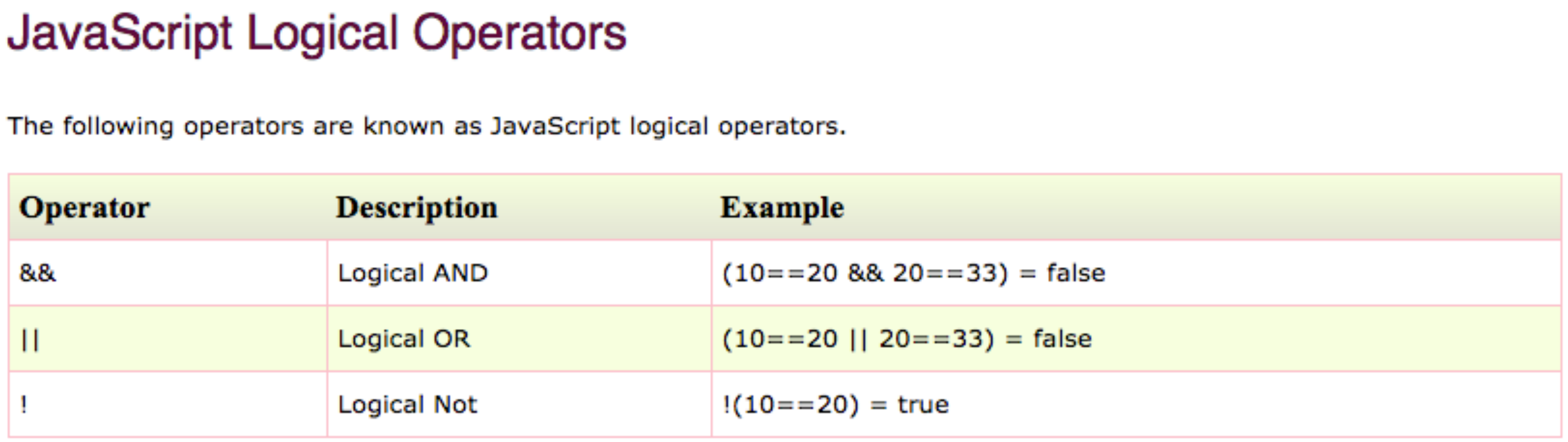


JavaScript Comparison Operators

The JavaScript comparison operator compares the two operands. The comparison operators are as follows:

Operator	Description	Example
==	Is equal to	10==20 = false
===	Identical (equal and of same type)	10==20 = false
!=	Not equal to	10!=20 = true
!==	Not Identical	20!==20 = false
>	Greater than	20>10 = true
>=	Greater than or equal to	20>=10 = true
<	Less than	20<10 = false
<=	Less than or equal to	20<=10 = false

JavaScript Operators

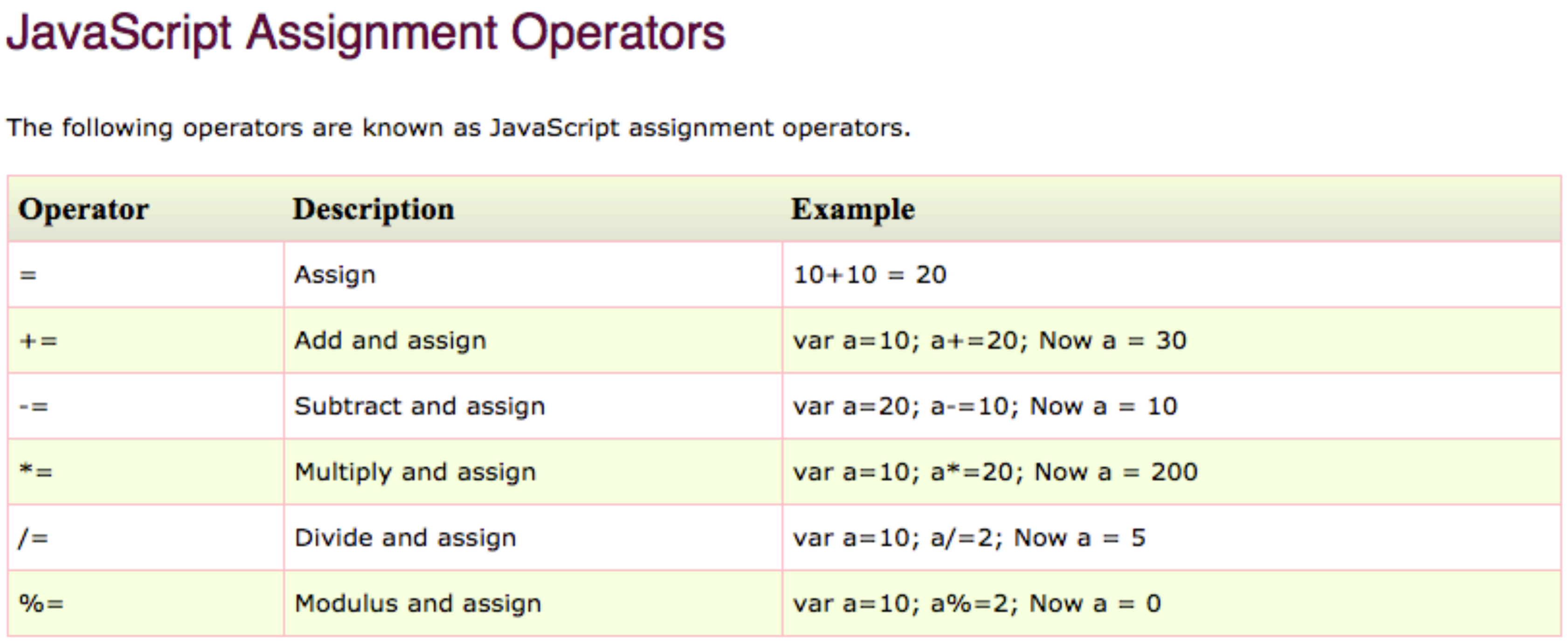


JavaScript Logical Operators

The following operators are known as JavaScript logical operators.

Operator	Description	Example
&&	Logical AND	(10==20 && 20==33) = false
||	Logical OR	(10==20 || 20==33) = false
!	Logical Not	!(10==20) = true

JavaScript Operators



JavaScript Assignment Operators

The following operators are known as JavaScript assignment operators.

Operator	Description	Example
=	Assign	10+10 = 20
+=	Add and assign	var a=10; a+=20; Now a = 30
-=	Subtract and assign	var a=20; a-=10; Now a = 10
*=	Multiply and assign	var a=10; a*=20; Now a = 200
/=	Divide and assign	var a=10; a/=2; Now a = 5
%=	Modulus and assign	var a=10; a%=2; Now a = 0

JavaScript Conditional

```
<script>
var a=20;
if(a>10){
    document.write("value of a is   greater than 10");
}
</script>

<script>
var a=20;
if(a%2==0){
    document.write("a is even number");
}
else{
    document.write("a is odd number");
}
</script>
```

JavaScript Conditional

```
<script>
var a=20;
if(a==10){
    document.write("a is equal to 10");
}
else if(a==15){
    document.write("a is equal to 15");
}
else if(a==20){
    document.write("a is equal to 20");
}
else{
    document.write("a is not equal to 10, 15 or 20");
}
</script>
```

JavaScript Conditional

```
<script>
var grade='B';
var result;
switch(grade){
     case 'A':
          result+=" A Grade";
          break;
     case 'B':
          result+=" B Grade";
          break;
     case 'C':
          result+=" C Grade";
          break;
     default:
          result+=" No Grade";
}
document.write(result);
</script>
```

JavaScript Conditional

```
<script>
var grade='B';
var result;
switch(grade){
     case 'A':
          result+=" A Grade";
          break;
     case 'B':
          result+=" B Grade";
          break;
     case 'C':
          result+=" C Grade";
          break;
     default:
          result+=" No Grade";
}
document.write(result);
</script>
```

JavaScript Loops

```
<script>
var i=21;
do{
   document.write(i + "<br/>");
   i++;
}while (i<=25);
</script>
```

JavaScript Function

```
<script>
function msg(){
    alert("hello! this is message");
}
</script>
<input type="button" onclick="msg()" value="call function"/>

<script>
function getcube(number){
    lert(number*number*number);
}
</script>
<form>
<input type="button" value="click" onclick="getcube(4)"/>
</form>
```

JavaScript Function

```
<script>
function getInfo(){
   return "hello bangsacerdas! How r u?";
}
</script>
<script>
   document.write(getInfo());
</script>
```

JavaScript Windows Object

```
<script>
   function msg(){
      alert("Hello Alert Box");
   }
</script>
<input type="button" value="click" onclick="msg()"/>
```

JavaScript Windows Object

```
<script>
function msg(){
    var v= confirm("Are u sure?");
    if(v==true){
        alert("ok");
    }
    else{
        alert("cancel");
    }

}
</script>

<input type="button" value="delete record" onclick="msg()"/>
```

JavaScript Windows Object

```
<script>
function msg(){
    var v= confirm("Are u sure?");
    if(v==true){
        alert("ok");
    }
    else{
        alert("cancel");
    }

}
</script>

<input type="button" value="delete record" onclick="msg()"/>
```

JavaScript Windows Object

```
<script>
function msg(){
   open("https://www.bangsacerdas.com");
}
</script>
<input type="button" value="javatpoint" onclick="msg()"/>
```

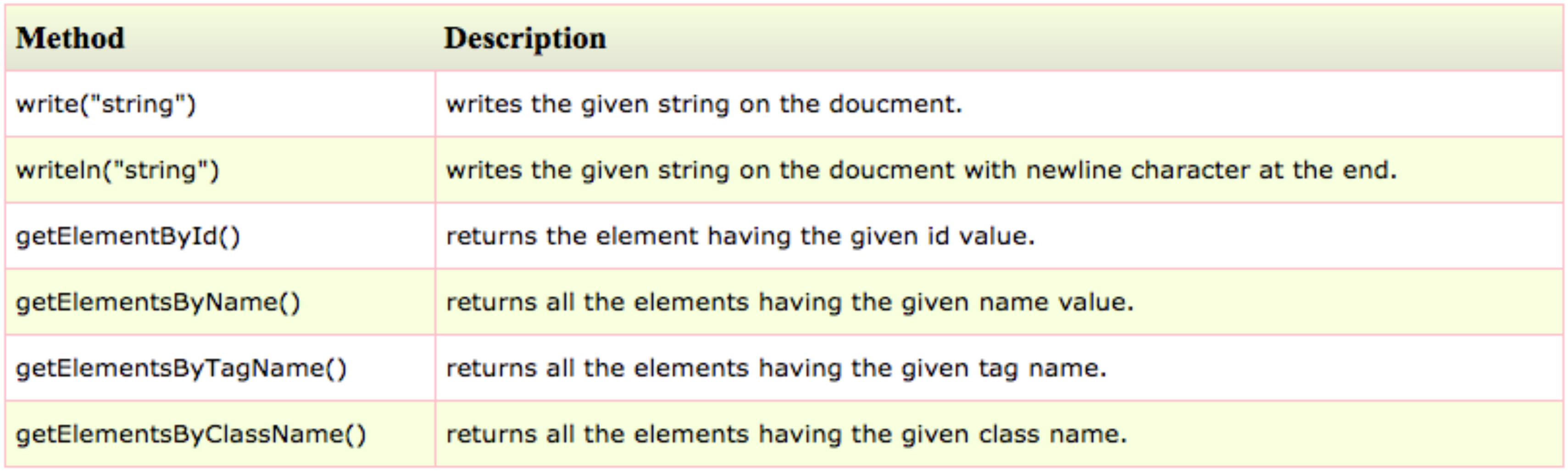
JavaScript Windows Object

```
<script>
function msg(){
setTimeout(
    function(){
    alert("Welcome to bangsacerdas after 2 seconds")
},2000);

}
</script>

<input type="button" value="click" onclick="msg()"/>
```

JavaScript DOM



Method	Description
write("string")	writes the given string on the doucment.
writeln("string")	writes the given string on the doucment with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

JavaScript DOM

```
<script type="text/javascript">
function printvalue(){
var name=document.form1.name.value;
alert("Welcome: "+name);
}
</script>

<form name="form1">
Enter Name:<input type="text" name="name"/>
<input type="button" onclick="printvalue()" value="print name"/>
</form>
```

JavaScript DOM

```
<script>
function getcube(){
    var number=document.getElementById("number").value;
    alert(number*number*number);
}
</script>

<form>
    Enter No:<input type="text" id="number" name="number”/<br/>
    <input type="button" value="cube" onclick="getcube()"/>
</form>
```

JavaScript DOM

```
<script type="text/javascript">
function totalelements()
{
    var allgenders=document.getElementsByName("gender");
    alert("Total Genders:"+allgenders.length);
}
</script>
<form>
    Male:<input type="radio" name="gender" value="male">
    Female:<input type="radio" name="gender" value="female">

    <input type="button" onclick="totalelements()”
value="Total Genders">
</form>
```

JavaScript DOM

```
<script type="text/javascript">
function countpara(){
var totalpara=document.getElementsByTagName("p");
alert("total p tags are: "+totalpara.length);

}
</script>
<p>This is a pragraph</p>
<p>Here we are going to count total number of paragraphs
by getElementByTagName() method.</p>
<p>Let's see the simple example</p>
<button onclick="countpara()">count paragraph</button>
```

JavaScript DOM

```
<script>
function showcommentform() {
    var data=”
    Name:<input type='text’name='name'>
    <br>Comment:<br><textarea rows='5' cols='80'></textarea>
    <br><input type='submit' value='Post Comment'>";
    document.getElementById('mylocation').innerHTML=data;
}
</script>
<form name="myForm">
    <input type="button" value="comment” onclick="showcommentform()">
    <div id="mylocation"></div>
</form>
```

JavaScript DOM

```
<script>
function validate() {
     var msg;
if(document.myForm.userPass.value.length>5){
     msg="good";
}
else{
     msg="poor";
}
document.getElementById('mylocation').innerText=msg;
 }

</script>
<form name="myForm">
<input type="password" value="" name="userPass" onkeyup="validate()">
Strength:<span id="mylocation">no strength</span>
</form>
```