HyperText Markup Language

HTML&CSS

STEP

IT ACADEMY

Unit 2

Style Sheet. Divs & Spans

Contents

Style Sheet

History of CSS Creation and Development

CSS is a special language that describes the appearance of a document created with the markup language. CSS is short for Cascading Style Sheets.

This term was proposed by the Norwegian engineer Håkon Wium Lie in 1994, the Chief Technology Officer of Opera Software at the moment.

The first version was adopted in 1996. It allowed changing font, color, and text attributes, aligning text, and so on.

In 1998 the CSS version 2 was released, which made block layout possible.

CSS 3 specification was published in 2006, animation and possibility of using variables appeared there.

Using CSS

Ways of Integrating CSS in pages

There are three main ways of integrating CSS tables in the html layout:

1. Use the `style` attribute.

```
<p style=”text-align:right”>...</p>
```

2. Use the `<style>` tag placed in the <head> tag.

```
<head>...
    <style type=”text/css”>...</style>
...</head>
```

3. Via separate file with the **.css** extension linked to a page.

CSS Semantics

Application Features

CSS semantics is quite simple:

```
Selector
{
Rule: value;
Rule: value;
Rule: value;
...
}
```

Tag name, as well as a separate class or identifier can be a selector.

Selector Expression

Selector – Tag Name

If tag name is specified as a selector, in this case all elements of this tag will conform to this style.

At that, if text elements are inside these tags, they inherit certain rules.

```
P
{
font-family:Segoe UI;
font-size:18pt;
colonGreen;
}
```

Selector – Class

Class is a special selector type which selects one or multiple elements.

To specify the class selector, you should use the .class name structure in CSS.

```
.class1
{
    font-family:Segoe Ul;
    font-size: 18pt;
    color:Green;
}
```

In order to bind an element to this class, specify the following attribute: class = “class name” in the opening tag

```
<p class=”classr’>Paragraph(j)</p>
```

Selector – Identifier

Identifier is a special selector type which selects one element. It is assumed that by default there is one element with this selector on a page.

```
...
#ldl
{
    font-family:Segoe Ul;
    font-size: 18pt;
    color:Green;
}
...
```

To specify the identifier selector, use the #identifier name structure in CSS.

```
<p id=”ld1”/>Paragraph</p>
```

To bind an element to this identifier, specify the id = “identifier name” attribute in the opening tag.

Selector Structures

Complex Selectors

There are several types of complex selectors:

1. Child selectors.
2. Multiclasses.
3. Sibling selectors.

Child Selectors

If you need to define style for an element nested in another element, the best decision would be the child selectors structure.

Selector of the parent element is specified first, and the second one is a child element selector nested in it

```
(parent_element) (child_ element)
{
…
}

<parent element>
<child element>…</child element>
</parent element>
```

Multiclasses

If you need to create style for an element, which is bound to several selectors, use the following structure:

```
selector, selector {}
```

All selectors, to which the element should conform, are listed without spaces in turn.

```
class 1.class2
{
…
}
…
<tag name class=”class1 class2”>
…
</tag name>
```

Sibling Selector

Sibling are layout elements that follow each other in the document code.

To set a style to the sibling element, use the + character, which is placed between two selectors.

```
selector1 + selector2
{
…
}
…
<selector1>... </selector1> <selector2>... </selector2>
```

Divs

The <div> tag Features

Concept of layer

The `<div> ... </div>` tag designed for layer creation is a container, where one can place almost any content. It varies in size and positioning, one can implement an agile enough layout.

```
<div>
    ...
    some container content
    ...
</div>
```

Having replaced the already obsolete tables, it is currently a main tool for page layout.

Advantages

Advantages of the div Layout

Layout using the `<div>` tag has certain advantages over table layout.

4. It doesn’t depend on dimensions of the neighboring elements.
5. You can implement the required level of nesting, which is not allowed in tables.
6. Layer layout loads faster.

Layer Size

How is Layer Size Set

Block size is usually set with the help of the corresponding `width` and `height` properties:

```
<div style=”width:200px; height:300px;
background-color:Blue”x/div>
```

Nesting

What Can be in a Layer

Any element can be a layer content:

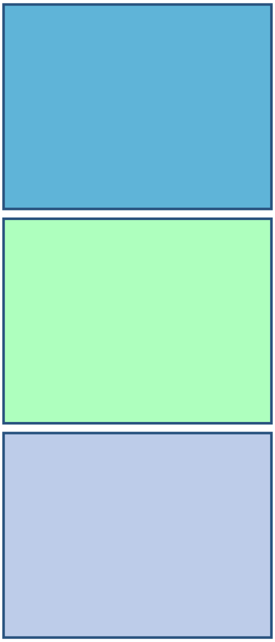
1. Table
2. List
3. Another layer
4. Any text content
5. Images

Layer Position

The <div> Elements Positioning

All layers are positioned one after another by default, each next one begins with a new row.

At that, layers are placed as close to the upper left corner of the page as possible.



Spaces from frames between layers are absent by default.

Vertical layer size is defined by content, and horizontal by the maximum available value according to the parent element size.

Wrapping of Elements

The float Property

In order to place elements horizontally, use the `float` property.

The `float` Property defines edge to align an element to and how it will be wrapped by other elements.

It can take 3 values:

1. `left` — the element is aligned left and wrapped from right.

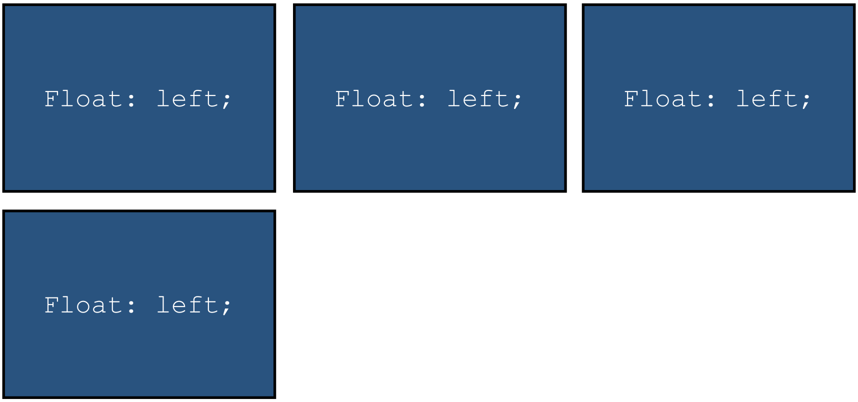
2. right — the element is aligned right and wrapped from left.
3. none — wrapping of element is not set.

Clear

Applying the clear Property

The clear property cancels wrapping or forbids wrapping an element from the specified side.

Property can take several values, but the most used one is both, it removes wrapping from both sides.

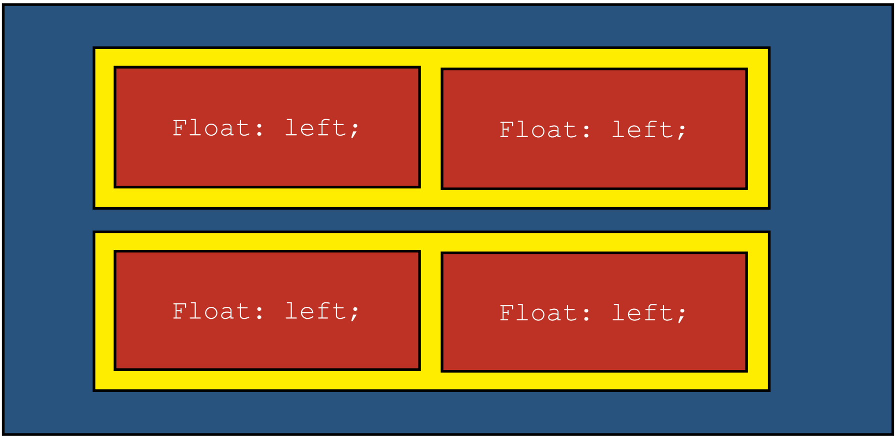


```
Float: left; I Float: left; I Float: left;
Clear: both;
```

Layer Nesting

Box Layout

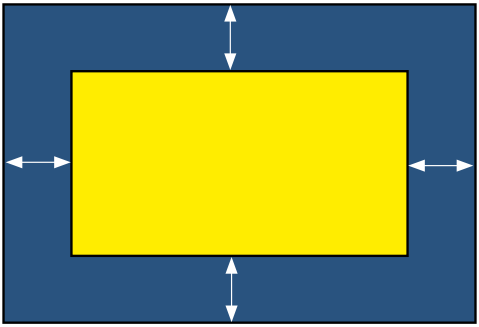
Unlike tables, here it is totally allowable to set layer nesting. At that, it should be initially thought through. A simple example is a square implementation.



Spaces

The padding Property

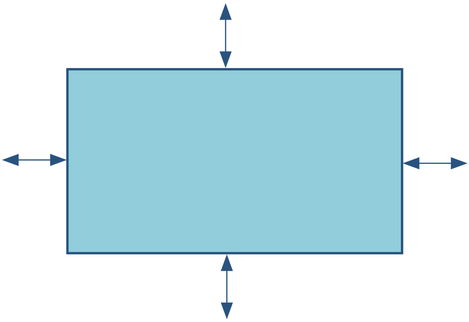
The padding property is responsible for inner space of an element, from its edges to content.



```
<div style=”padding: 30px;”>...</div>
```

The margin Property

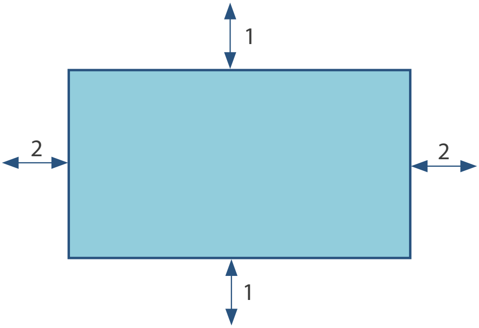
The margin property is responsible for outer space of an element, from its borders to borders of the neighbor or parent elements.



```
<div style=” margin:30px;”>...</div>
```

- **The** padding **and** margin **properties that take 2 parameters** If a block has one of these properties set and it is given 2 parameters, then the first of them is responsible for vertical margins, and the second one for horizontal margins.

```
<div style=”margin: Opx auto”>...</div>
```

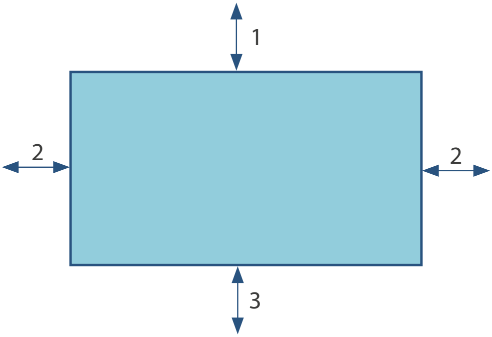


The auto value is used frequently.
It says that spaces should be equal from all sides.

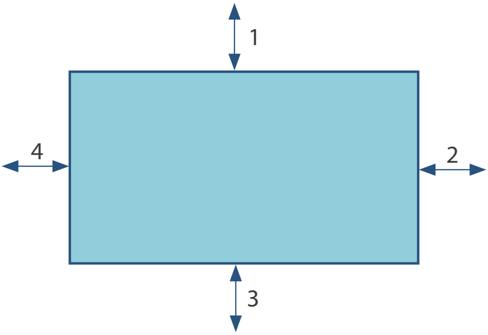
- **The** padding **and** margin **properties that take 3 parameters** If block has one of these properties set and it is given 3 parameters, then the first one sets margin on top, the second one on the left and on the right, and the third at the bottom.

```
<div style=”margin: 0px 20px 30px”>...</div>
```

It is worth remembering that the first parameter is always responsible for the top margin, and the last one for the left margin.



- **The** `padding` **and** `margin` **properties that take 4 parameters** In case when all four parameters are set, you need to remember the clockwise order: first one on top, second to the right, third at the bottom, fourth to the left.



```
<div style=”margin: 0px 20px 30px 40px”>...</div>
```

The <span>

Text Formatting

The <span>...</span> Tag

The <span> tag is designed for document text formatting. Unlike box elements, using the <span> tag, you can select a part of information inside other tags and set its own style for it.

```
<span style=”color:Red”> span </span>
```

The text specified inside of this element gains all properties described in the opening tag.

The <span> tag perceives margin, padding, width, and height properties differently in different browsers, so you have to use them carefully.

Unit 2.
Style Sheet. Divs & Spans