CSS Selectors

In CSS, selectors are patterns used to select the element(s) you want to style.

Try it <http://www.w3schools.com/cssref/trysel.asp>

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [.*class*](http://www.w3schools.com/cssref/sel_class.asp) | .intro | Selects all elements with class="intro" |
| [#*id*](http://www.w3schools.com/cssref/sel_id.asp) | #firstname | Selects the element with id="firstname" |
| [\*](http://www.w3schools.com/cssref/sel_all.asp) | \* | Selects all elements |
| [*element*](http://www.w3schools.com/cssref/sel_element.asp) | p | Selects all <p> elements |
| [*element,element*](http://www.w3schools.com/cssref/sel_element_comma.asp) | div, p | Selects all <div> elements and all <p> elements |
| [*element* *element*](http://www.w3schools.com/cssref/sel_element_element.asp) | div p | Selects all <p> elements inside <div> elements |
| [*element*>*element*](http://www.w3schools.com/cssref/sel_element_gt.asp) | div > p | Selects all <p> elements where the parent is a <div> element |
| [*element*+*element*](http://www.w3schools.com/cssref/sel_element_pluss.asp) | div + p | Selects all <p> elements that are placed immediately after <div> elements |
| [*element1*~*element2*](http://www.w3schools.com/cssref/sel_gen_sibling.asp) | p ~ ul | Selects every <ul> element that are preceded by a <p> element |
| [[*attribute*]](http://www.w3schools.com/cssref/sel_attribute.asp) | [target] | Selects all elements with a target attribute |
| [[*attribute*=*value*]](http://www.w3schools.com/cssref/sel_attribute_value.asp) | [target=\_blank] | Selects all elements with target="\_blank" |
| [[*attribute*~=*value*]](http://www.w3schools.com/cssref/sel_attribute_value_contains.asp) | [title~=flower] | Selects all elements with a title attribute containing the word "flower" |
| [[*attribute*|=*value*]](http://www.w3schools.com/cssref/sel_attribute_value_lang.asp) | [lang|=en] | Selects all elements with a lang attribute value starting with "en" |
| [[*attribute*^=*value*]](http://www.w3schools.com/cssref/sel_attr_begin.asp) | a[href^="https"] | Selects every <a> element whose href attribute value begins with "https" |
| [[*attribute*$=*value*]](http://www.w3schools.com/cssref/sel_attr_end.asp) | a[href$=".pdf"] | Selects every <a> element whose href attribute value ends with ".pdf" |
| [[*attribute*\*=*value*]](http://www.w3schools.com/cssref/sel_attr_contain.asp) | a[href\*="w3schools"] | Selects every <a> element whose href attribute value contains the substring "w3schools" |

media Rule

## Definition and Usage

The @media rule is used to define different style rules for different media types/devices.

In CSS2 this was called media types, while in CSS3 it is called media queries.

Media queries look at the capability of the device, and can be used to check many things, such as:

* width and height of the viewport
* width and height of the device
* orientation (is the tablet/phone in landscape or portrait mode?)
* resolution
* and much more

Try it <http://www.w3schools.com/cssref/css3_pr_mediaquery.asp>

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

.wrapper {overflow:auto;}

#main {margin-left: 4px;}

#leftsidebar {float: none;width: auto;}

#menulist {margin:0;padding:0;}

.menuitem {

background:#CDF0F6;

border:1px solid #d4d4d4;

border-radius:4px;

list-style-type:none;

margin:4px;

padding:2px;

}

@media screen and (min-width: 480px) {

#leftsidebar {width:200px;float:left;}

#main {margin-left:216px;}

}

</style>

</head>

<body>

<div class="wrapper">

<div id="leftsidebar">

<ul id="menulist">

<li class="menuitem">Menu-item 1</li>

<li class="menuitem">Menu-item 2</li>

<li class="menuitem">Menu-item 3</li>

<li class="menuitem">Menu-item 4</li>

<li class="menuitem">Menu-item 5</li>

</ul>

</div>

<div id="main">

<h1>Resize the browser window to see the effect!</h1>

<p>This example shows a menu that will float to the left of the page if the viewport is 480 pixels wide or wider. If the viewport is less than 480 pixels, the menu will be on top of the content.</p>

</div>

</div>

</body>

</html>

# **CSS display Property**

The display property specifies the type of box used for an HTML element.

The display property defines the type of element box an element generates

in the layout. In addition to the familiar inline and block display roles, you

can also make elements display as list items or the various parts of a table.

As you can see from the list of values, there are a lot of roles an element can

play, but there are only a handful that are used in everyday practice.Try it <http://www.w3schools.com/cssref/playit.asp?filename=playcss_display&preval=inline>

# **CSS position Property**

specifies which positioning method to use. I’ll introduce each keyword value

briefly here, and then we’ll take a more detailed look at each method in the

remainder of this chapter.

* static

This is the normal positioning scheme in which elements are positioned

as they occur in the normal document flow.

* relative

Relative positioning moves the box relative to its original position in the

flow. The distinctive behavior of relative positioning is that the space

the element would have occupied in the normal flow is preserved as an

empty space.

* absolute

Absolutely positioned elements are removed from the document flow

entirely and positioned with respect to the browser window or a containing

element (we’ll talk more about this later). Unlike relatively positioned

elements, the space they would have occupied is closed up. In fact, they

have no influence at all on the layout of surrounding elements.

* fixed

The distinguishing characteristic of fixed positioning is that the element

stays in one position in the window even when the document scrolls.

Fixed elements are removed from the document flow and positioned

relative to the browser window (or other viewport) rather than another

element in the document. It currently causes some hiccups on mobile

devices, as discussed later in this chapter.

Each positioning method has its purpose, but absolute positioning is the

most versatile. With absolute positioning, you can place an object anywhere

in the viewport or within another element. Absolute positioning can even be

used to create multicolumn layouts, but it is more commonly used for small

tasks, like positioning a search box in the top corner of a header. You can

also use absolute positioning to break an image or chunk out of its containing

box, creating hanging indents or overlap effects. It’s a handy tool when

used carefully and sparingly.

Try it <http://www.w3schools.com/cssref/pr_class_position.asp>

# **CSS float Property**

the float property moves an element as far as possible to the left or right, allowing the following content to wrap around it

<http://www.w3schools.com/cssref/playit.asp?filename=playcss_float&preval=none>

# **CSS z-index Property**

The value of the z-index property is a number (positive or negative). The

higher the number, the higher the element will appear in the stack. Lower

numbers and negative values move the element lower in the stack. Let’s look

at an example to make this clear

<http://www.w3schools.com/cssref/playit.asp?filename=playcss_z-index>

Practice

Create web page by using resource in folder practice that given html and images file.

1.You must use style property that we learn today (selector, media rules, position float, z-index)

2.You can use any element and style you like.

To transmit file

1.Create Folder INT106 in desktop

2.zip your work folder as w\_student id (ex. w\_58130500000.zip)

3.copy the zip file to folder INT106