

Table of Contents

1 Semantic HTML

2 CSS Selectors

3 Background

4 Media Queries

5 Transitions

5.1 Keyframes . . . . .

6 Gradients

6.1 Examples . . . . .

7 Flexbox

7.1 Parent . . . . .

7.2 Client . . . . .

8 Borders

9 Box Model

10 Code Snippets

10.1 Example Page Layout . . . . .

10.2 Details . . . . .

the rest of the site

**Aside:** Content placed to the side, related to surrounding content

**Details:** Additional details the user can show or hide on demand

**FigCaption:** Caption for an image

**Figure:** Defines an image, typically includes a figcaption

**Footer:** Footer of a document or section. Typically: Author, Copyright Information, Links to Terms of Use, Contact Information, etc

**Header:** Header of a document or section

**Main:** Main content of the document

**Mark:** Highlights part of the text

**Nav:** Block for major navigational links

**Section:** “A section is a thematic grouping of content, typically with a heading”

**Summary:** Defines a visible heading for the details tag

**Time:** Defines a human readable date/times

1 Semantic HTML

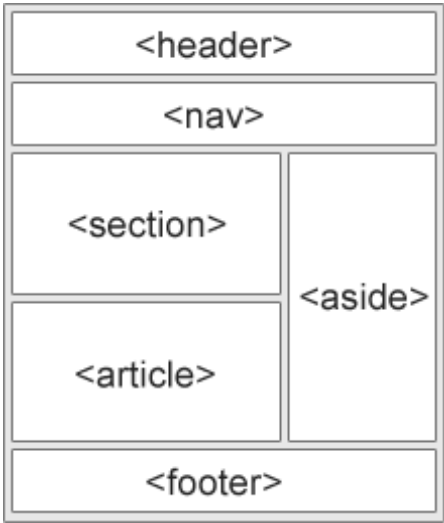


Figure 1: Basic breakdown of semantic html (source: w3schools)

**Article:** Makes sense on it’s own, the content should be understandable regardless of

2 CSS Selectors

**.class** Selects all elements with class equal to “class”

**#id** Selects all elements with id equal to “id”

**\*** Selects all elements

**element** Selects all elements by tag name

**element,element** Multiple css selectors

**element element** Selects all elements inside an element

**element > element** Selects all direct children of an element

**element + element** Selects all elements directly after an element

**element element** Selects all elements preceded by an element

**[attribute]** Selects elements with the attribute

**[attribute=value]** Selects elements with attribute equal to value

**[attribute =value]** Selects elements with attribute containing value

**[attribute|=value]** Selects elements with attribute starting with value

**[attribute^=value]** Selects elements with attribute ending with value

**[attribute\*=value]** Selects elements with

attribute containing substring value

**:active** Selects the active link  
**::after** Insert something after the content  
**::before** Insert something before the content  
**:checked** Selects every checked input  
**:default** Selects the default input  
**:disabled** Selects every disabled input  
**:empty** Selects parents with no children  
**:enabled** Selects every enabled input  
**:first-child** Selects if first child of parent  
**::first-letter** Selects the first letter  
**::first-line** Selects the first line  
**:first-of-type** Selects if first element of parent  
**:focus** Selects input which has focus  
**:hover** Selects if mouse is over  
**:in-range** Selects input elements with a value specified within a range  
**:indeterminate** Selects input elements that are indeterminate  
**:invalid** Selects all input elements with an invalid value  
**:lang(language)** Selects with lang attribute equal to language  
**:last-child** Selects elements that are the last child of the parent  
**:last-of-type** Selects elements that are the last child of the parent  
**:link** Selects all unvisited links  
**:not(selector)** Selects every element that is not the element  
**:nth-child(n)** Selects every element that is the nth child of the parent  
**:nth-last-child(n)** Selects every element that is the last nth child of the parent  
**:nth-last-of-type(n)** Selects every element that is the last nth child of the parent  
**:nth-of-type(n)** Selects every element that is the nth of type from the parent  
**:only-of-type** Selects every element that is the only type from parent  
**:only-child** Selects every element that is the only child of its parent  
**:optional** Selects inputs with no required attributes  
**:out-of-range** Selects input elements with a value outside the specified range  
**::placeholder** Selects inputs with a placeholder text  
**:read-only** Selects inputs with the readonly attribute

**:read-write** Selects inputs without the readonly attribute

**:required** Selects inputs with the required attribute

**:root** Selects the document's root element

**::selection** Selects the portion of an element that is selected by a user

**:target** Selects the current active element (clicked on a URL containing that anchor name)

**:valid** Selects all input elements with a valid value

**:visited** Selects all visited links

## 3 Background

---

**background-image:** selects the background image to use, accepted value takes the form `url("image.jpg")`

**background-color:** specifies the background color in hex

**background-position:** positions the image, accepted values:

- (left | right | center) (top | bottom | center)
- x% y%
- x y

Position in px

**background-repeat:** enables the image to be repeated, possible values:

- repeat
  - repeat-x
  - repeat-y
  - no-repeat
  - space
- repeat without clipping and remaining space distributed evenly
- round
- repeat without clipping and centered

**background-size:** specifies the size of the background images, possible values:

- auto
- displayed at original size
- *height width*
- sets the image height and width in pixels, percentage is also supported
- cover

resize the image to cover the entire screen (stretches or cuts off the sides if needed)

- contain  
resize to make the entire image visible

## 4 Media Queries

- @media screen - executes only if viewed on a screen
- @media print - executes when about to print
- @media all and (max-width: 699px) and (min-width: 520px) - executes when the display has a width of 520 to 699px

## 5 Transitions

**transition:** <attribute> <duration>

**transition-delay:** delay before the effect starts (e.g. 2s)

**transition-duration:** duration of the transition

**transition-property:** attribute to transition on

**transition-timing-function:** possible values: ease, linear, ease-in, ease-out, ease-in-out, cubic-bezier(n, n, n, n)

### 5.1 Keyframes

```
1 @keyframes mymove {
2     0% {
3         top: 0px;
4     }
5     100% {
6         top: 200px;
7     }
8 }
9
10 div {
11     animation: mymove 2s infinite;
12 }
```

**animation:** <name> <duration> <iteration>

**animation-direction:** (reverse | alternate | alternate-reverse)

**animation-fill-mode:** (none | forwards |

backwards | both)

Iteration can be a number or infinite

## 6 Gradients

linear-gradient(<position>, <color1>, <color2>)

radial-gradient(<color1>, <color2>)

### 6.1 Examples

background-image: linear-gradient(-90deg, red, yellow);

background-image: radial-gradient(circle, red, yellow);

## 7 Flexbox

### 7.1 Parent

display: flex;

**flex-direction:** row | row-reverse | column | column-reverse

*Sets flexbox direction*

**flex-wrap:** nowrap | wrap | wrap-reverse

*Sets if the flexbox will wrap or not*

**justify-content:** flex-start | flex-end | center | space-between | space-around | space-evenly

*Aligns the items across the wrap axis*

**align-items:** flex-start | flex-end | center | stretch | baseline

*Aligns the items across the **cross** wrap line*

**align-content:** flex-start | flex-end | center | stretch | space-between | space-around

*Aligns all wrapped rows inside the parent*

### 7.2 Client

**order:** <integer>

*Position the individual flex item within the parent*

**flex-grow:** <number>

*Specifies if the item should grow and the rate it should grow*

**flex-shrink:** <number>

*Specifies if the item should shrink and the rate it will shrink*

**flex-basis:** <length> | auto

*Specify the default item width*

**align-self:** auto | flex-start | flex-end | center | baseline | stretch

*Overrides the parent align-items for the individual item*

## 8 Borders

**border-style:** dotted | dashed | solid | double | groove | ridge | inset | outset | none | hidden

*Supports specific edges as well:*  
border-top-style

**border-width:** <number>

**border-color:** <color>

**border-radius:** <size>

*Makes rounded corners*

## 9 Box Model

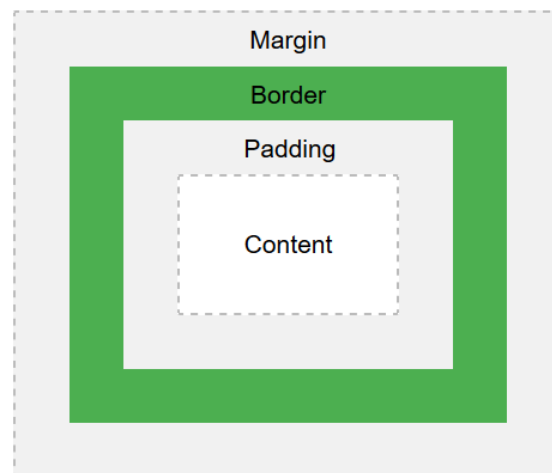


Figure 2: Layout of boxmodel (taken from w3schools)

### 9.1 box-sizing

**content-box:** Width and height only includes content (border and padding not included)

**border-box:** Width and height includes content, padding and border

## 10 Code Snippets

### 10.1 Example Page Layout

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="utf-8" />
5     <title>Page Title</title>
6     <style>
7         :root {
8             --primary: #482f40;
9         }
10    </style>
11    <link rel="stylesheet" href="styles.css" />
12 </head>
13 <body>
14     
15
16     <script type="text/javascript" src="scripts.js"></script>
17 </body>

```

### 10.2 Details

```
1 <details>
2     <summary>World's Tallest Mountain</summary>
3     <p>Here you would place some information about the worlds tallest moon</p>
4 </details>
```