file structures and URLS

- absolute
- relative

More CSS

- specificity
- file directories
  children, parents, grandparents and siblings

CSS

Specificity,
Inheritance and
Cascade
(and multiple classes)

http://www.vanseodesign.com/css/css-specificity-inheritance-cascaade/

clear float
clear fix

Inheritance and Cascade

If the rules are equal in specificity individual rules get overridden **in the order they're defined in the CSS**, so in your example red wins because it comes later in the CSS definitions. The same rule applies in other cases as well, for example:

```
<div class="red green">
```

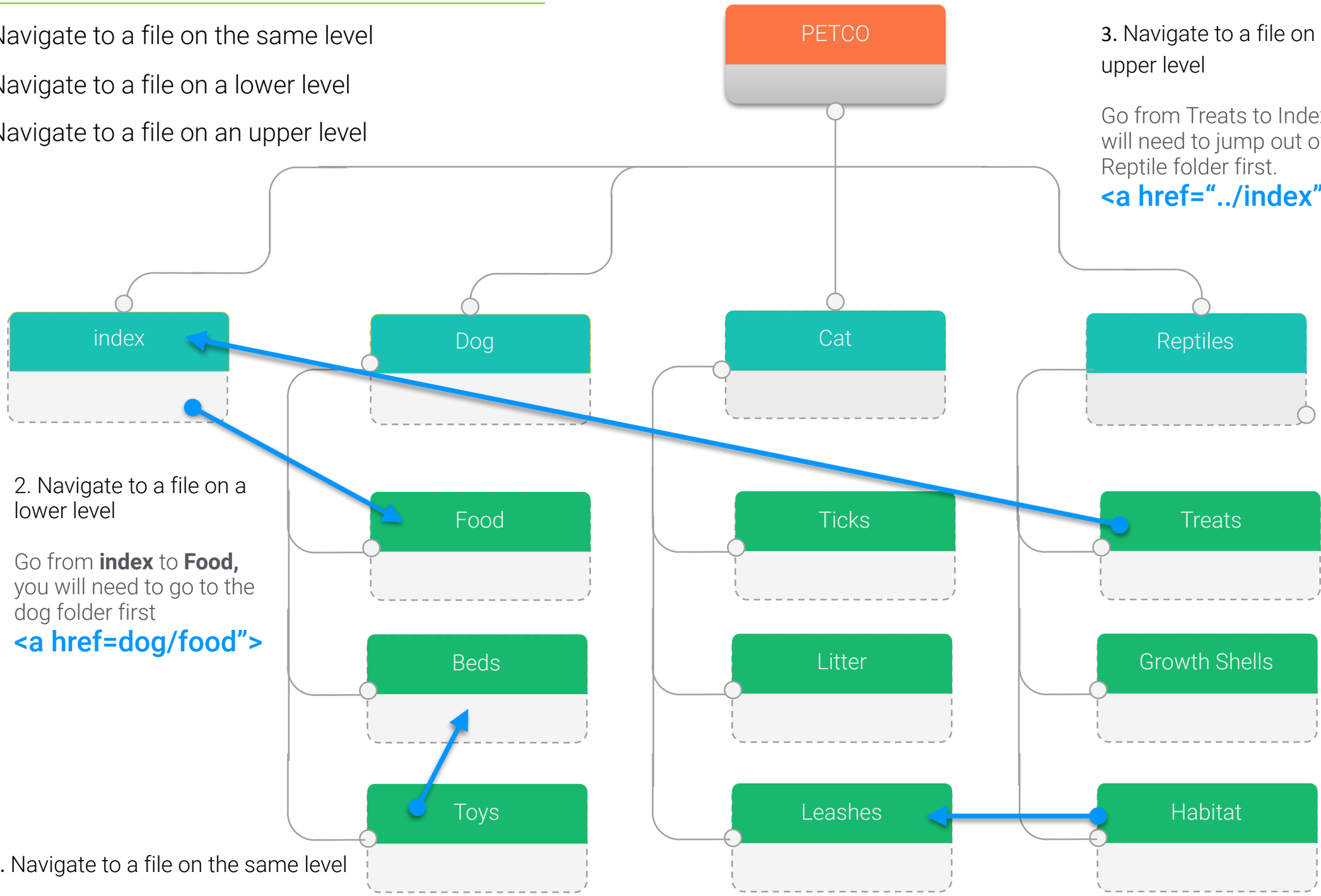
Which of these wins?

```
.green { color: green; }
.red { color: red; }
```

`.red` wins here, it doesn't matter the order in the `class` attribute, all that matters is the order the styles are defined in the CSS itself.

3 types of relative paths

1. Navigate to a file on the same level
2. Navigate to a file on a lower level
3. Navigate to a file on an upper level



3. Navigate to a file on an upper level

Go from Treats to Index, you will need to jump out of the Reptile folder first.

<a href="../index">

2. Navigate to a file on a lower level

Go from **index** to **Food,** you will need to go to the dog folder first

<a href=dog/food">

1. Navigate to a file on the same level

Go from **Toys** to **Beds**

<a href="toys">

Another Example: If you are on **Habitat** and you want to go to **Leashes:**

<a href=../cat/leashes">

ABSOLUTE URLS

When you link to a different website, the value of the `href` attribute will be the full web address for the site, which is known as an **absolute** URL.

ABSOLUTE URLS

URL stands for Uniform Resource Locator. Every web page has its own URL. This is the web address that you would type into a browser if you wanted to visit that specific page.

An absolute URL starts with the domain name for that site, and can be followed by the path to a specific page. If no page is specified, the site will display the homepage.

http://www.google.com

RELATIVE URLS

When linking to other pages within the same site, you can use relative URLs. These are like a shorthand version of absolute URLs because you do not need to specify the domain name.

RELATIVE URLS

HTML chapter-04/linking-to-other-pages.html

```
<p>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="about-us.html">About</a></li>
    <li><a href="movies.html">Movies</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</p>
```

<a>

When you are linking to other pages within the same site, you do not need to specify the domain name in the URL. You can use a shorthand known as a **relative** URL.

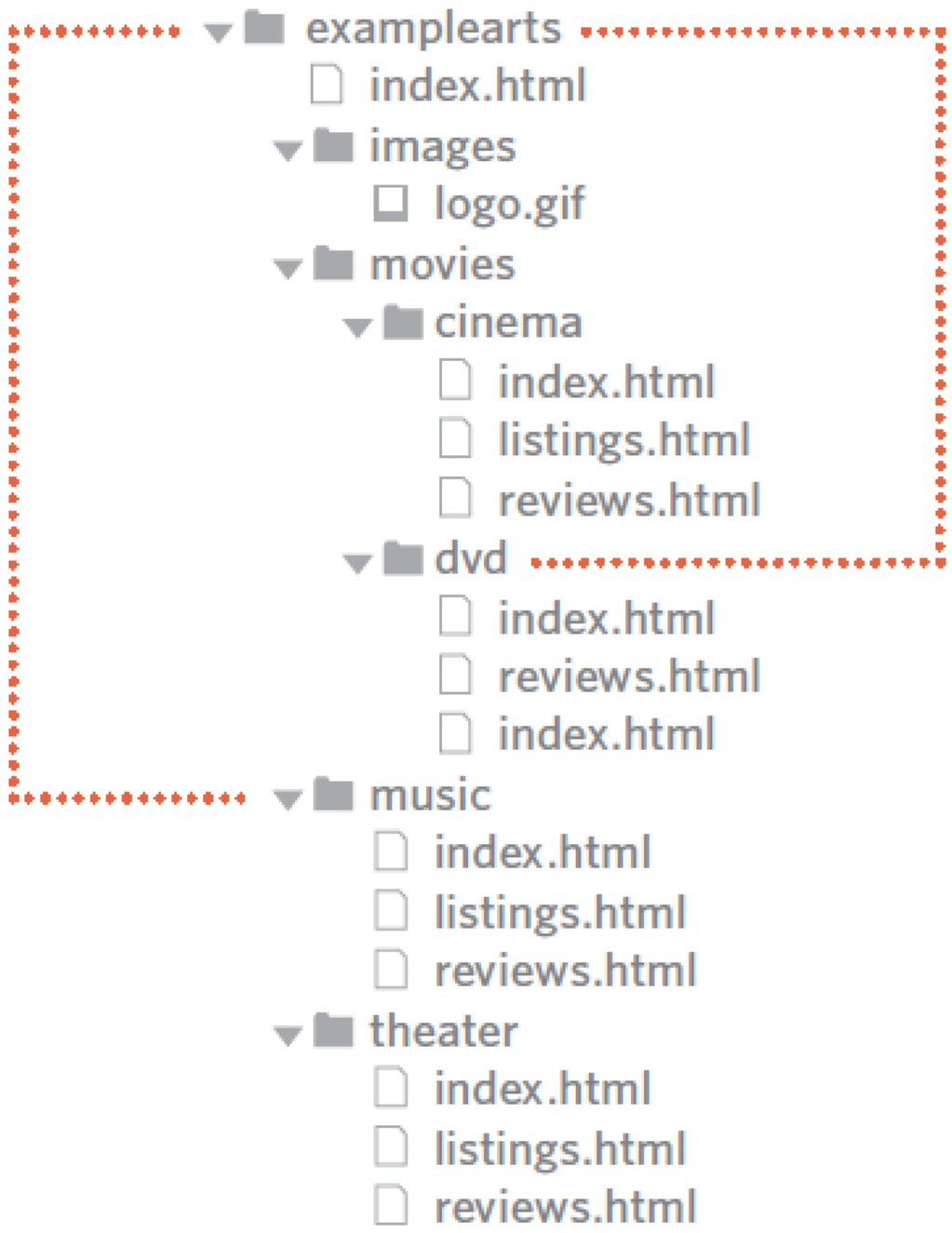
If all the pages of the site are in the same folder, then the value of the `href` attribute is just the name of the file.

PARENT

The *examplearts* folder is a parent of the *music* folder.

CHILD

The *music* folder is a child of the *examplearts* folder.



GRANDPARENT

The *examplearts* folder is a grandparent of the *dvd* folder.

GRANDCHILD

The *dvd* folder is a grandchild of the *examplearts* folder.

STRUCTURE

The diagram on the right shows the directory structure for a fictional entertainment listings website called ExampleArts.

The top-level folder is known as the **root** folder. (In this example, the root folder is called ***examplearts***.) The root folder contains all of the other files and folders for a website.

Each section of the site is placed in a separate folder; this helps organize the files.

RELATIONSHIPS

The relationship between files and folders on a website is described using the same terminology as a family tree.

In the diagram on the right, you can see some relationships have been drawn in.

The ***examplearts*** folder is a parent of the ***movies***, ***music*** and ***theater*** folders. And the the ***movies***, ***music*** and ***theater*** folders are children of the ***examplearts*** folder.

HOMEPAGES

The main homepage of a site written in HTML (and the homepages of each section in a child folder) is called ***index.html***.

Web servers are usually set up to return the ***index.html*** file if no file name is specified.

Therefore, if you enter `examplearts.com` it will return `examplearts.com/index.html`, and `examplearts.com/music` will return `examplearts.com/music/index.html`.

Every page and every image on a website has a **URL** (or Uniform Resource Locator). The URL is made up of the domain name followed by the **path** to that page or image.

The path to the homepage of this site is `www.examplearts.com/index.html`. The path to the logo for the site is `examplearts.com/images/logo.gif`.

You use URLs when linking to other web pages and when including images in your own site. On the next page, you will meet a shorthand way to link to files on your own site.

The root folder contains:

- A file called *index.html* which is the homepage for the entire site
- Individual folders for the movies, music and theatre sections of the site

Each sub-directory contains:

- A file called *index.html* which is the homepage for that section
- A reviews page called *reviews.html*
- A listings page called *listings.html* (except for the DVD section)

The movies section contains:

- A folder called *cinema*
- A folder called *DVD*.

If all of the files in your site are in one folder, you simply use the file name for that page.

If your site is organized into separate folders (or directories), you need to tell the browser how to get from the page it is *currently on* to the page that you are *linking to*.

If you link to the same page from two different pages you might, therefore, need to write two different relative URLs.

These links make use of the same terminology (borrowed from that of family trees) you met on the previous page which introduces directory structure.

RELATIVE LINK TYPE	EXAMPLE (from diagram on previous page)
SAME FOLDER To link to a file in the same folder, just use the file name. (Nothing else is needed.)	To link to music reviews from the music homepage: `<a href="reviews.html">Reviews</a>`
CHILD FOLDER For a child folder, use the name of the child folder, followed by a forward slash, then the file name.	To link to music listings from the homepage: `<a href="music/listings.html">Listings</a>`

GRANDCHILD FOLDER
Use the name of the child folder, followed by a forward slash, then the name of the grandchild folder, followed by another forward slash, then the file name.
	To link to DVD reviews from the homepage:
	`<a href="movies/dvd/reviews.html">`
	`Reviews</a>`

PARENT FOLDER Use `../` to indicate the folder above the current one, then follow it with the file name.	To link to the homepage from the music reviews: `<a href="../index.html">Home</a>`
GRANDPARENT FOLDER Repeat the `../` to indicate that you want to go up two folders (rather than one), then follow it with the file name.	To link to the homepage from the DVD reviews: `<a href="../../index.html">Home</a>`

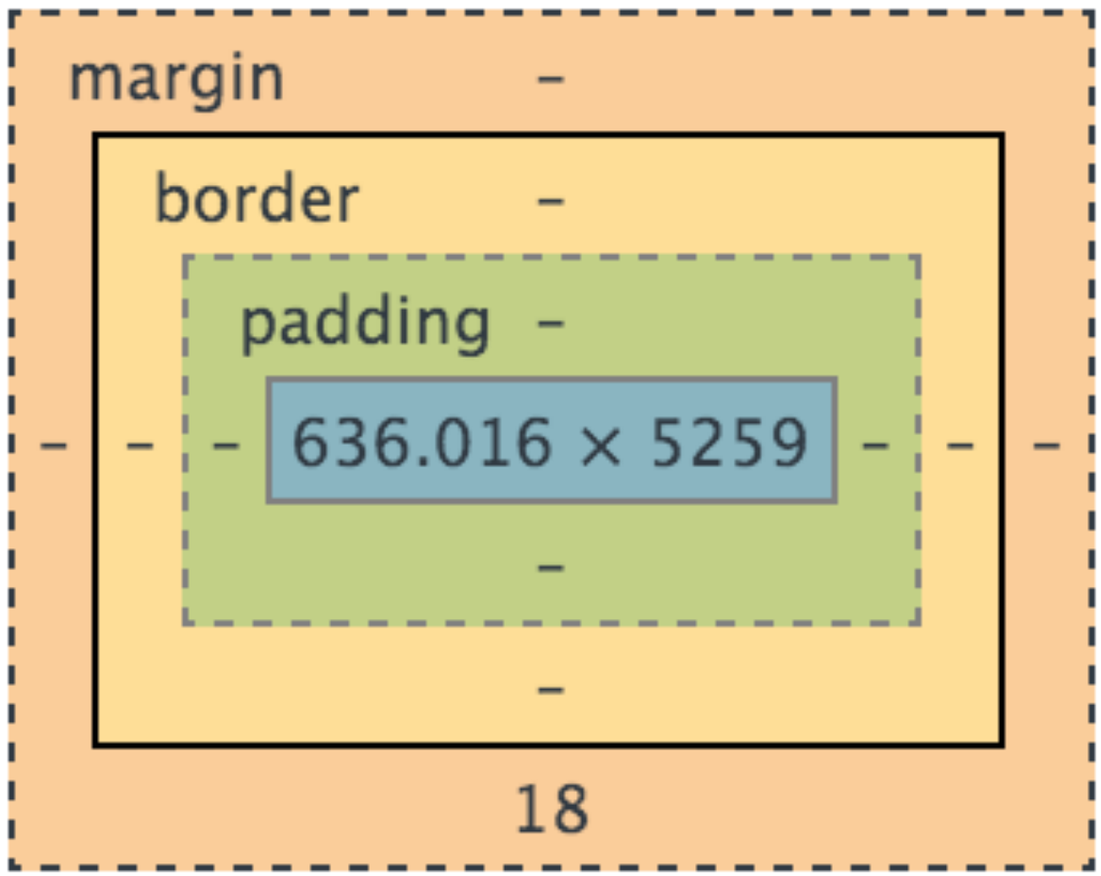
This lets us know that we require a

easily removable

A forward slash will return the homepage for the entire site, and a forward slash followed by a file name will return that file providing it is in the root directory.

The Box Model

- padding
- border
- margins

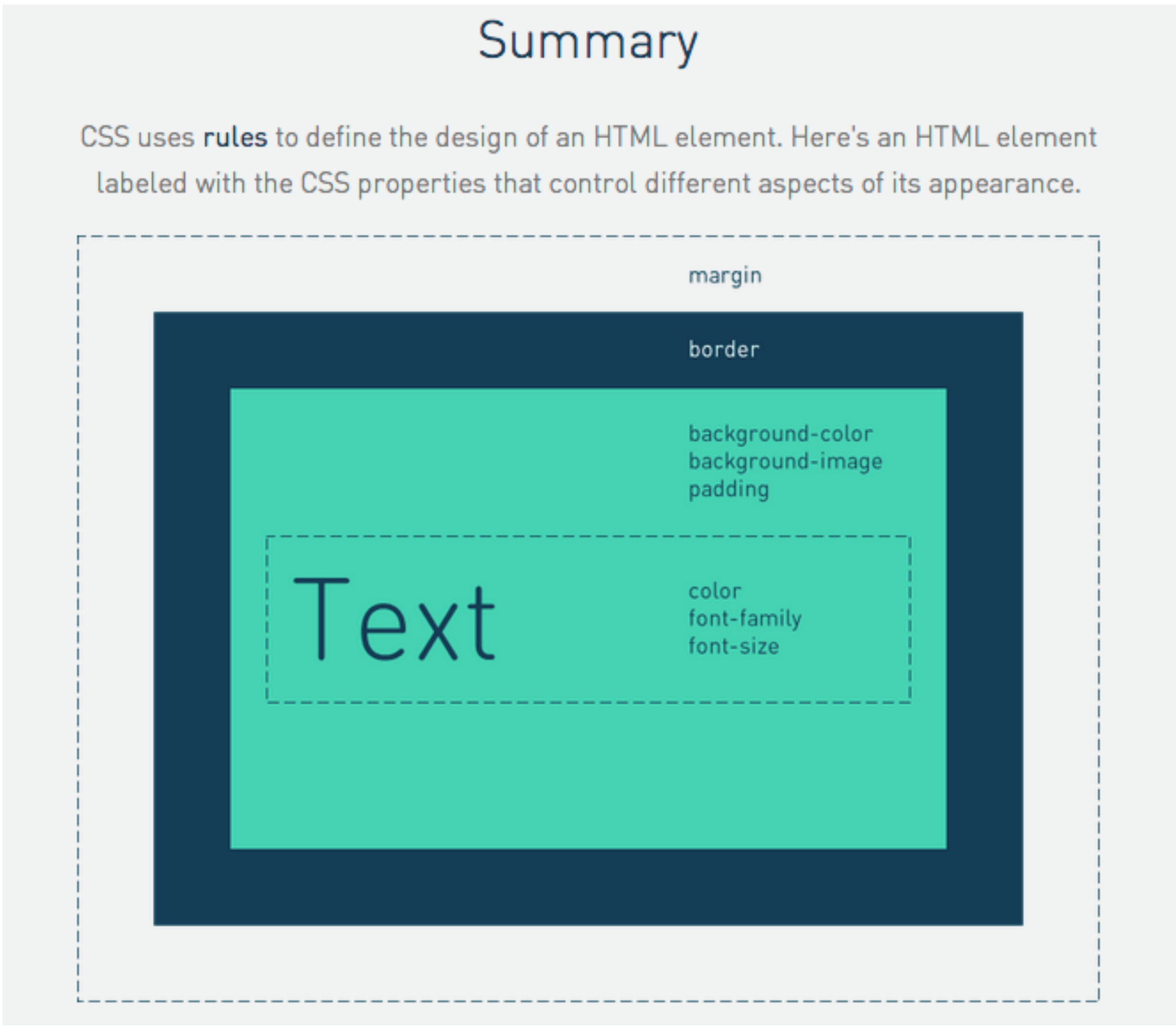


CSS

Box Model

Each HTML element is like a tiny box or container that holds the pictures and text you specify.

We've checked out the image in the browser tools: that's what the box model looks like! (We'll cover the details of margins, borders, and padding in the next section.)



CSS

Padding and Margins

When specifying padding or margins it starts from the upper right and circles to the lower left.

```
div {
  padding-top:30px;
}
.persian {
  padding: 10px 20px;
}

.container {
  margin: 10px 0 20px;
}
```

Understanding padding and margins is fundamental to using CSS. Practices such as using the height to create padding or margins further leads to bugs and inconsistencies. If you add padding to an inline element you get nothing in height.

PADDING is the inner space of an element
MARGIN is the outer space of an element.
The difference becomes clear once you apply backgrounds and borders to an element.

What background-color can you make margin?

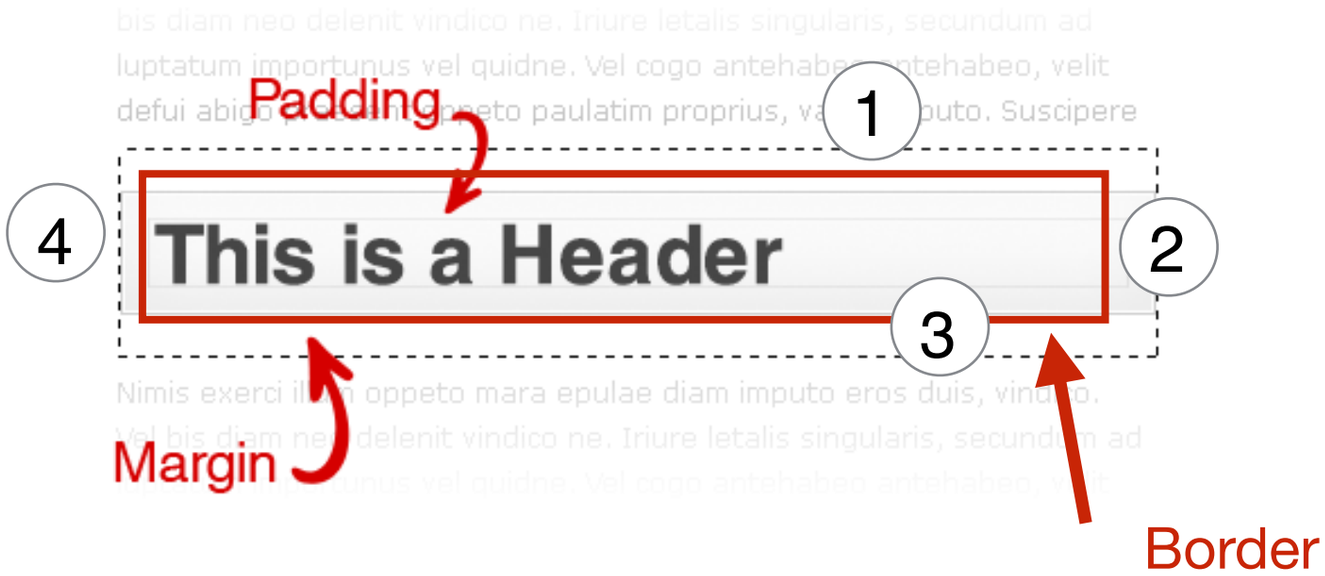
CSS

Border

Shorthand property The border property is a shorthand for the following individual border properties:

border-width
border-style (required)
border-color

```
p {
    border: 5px solid red;
}
p.one {
    border-style: solid;
    border-color: red;
border-width: 5px;
}
```



Border Properties

none: Defines no border
dotted: Defines a dotted border
dashed: Defines a dashed border
solid: Defines a solid border
double: Defines two borders. The width of the two borders are the same as the border-width value
groove: Defines a 3D grooved border. The effect depends on the border-color value
ridge: Defines a 3D ridged border. The effect depends on the border-color value
inset: Defines a 3D inset border. The effect depends on the border-color value
outset: Defines a 3D outset border. The effect depends on the border-color

Elements

Inline and Block

Each HTML element is like a tiny box or container that holds the pictures and text you specify.

These are block elements

<p>
<h1-h6>

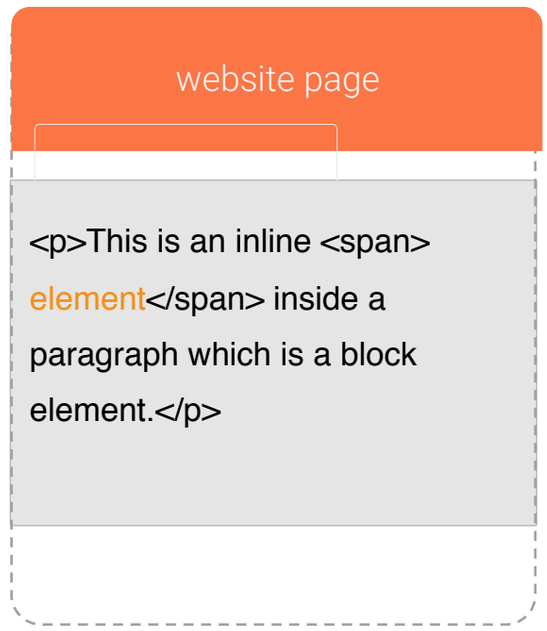
<form>

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline element inside a paragraph.

Examples of inline elements:

<a>
<img>



CSS

Specificity, Inheritance and Cascade
(and multiple classes)

http://www.vanseodesign.com/css/css-specificity-inheritance-cascaade/

coming up…
float and clear fix
CSS reset

Inheritance and Cascade

If the rules are equal in specificity individual rules get overridden **in the order they're defined in the CSS**, so in your example red wins because it comes later in the CSS definitions.
Here we are applying multiple classes:

```
<div class="red green">
```

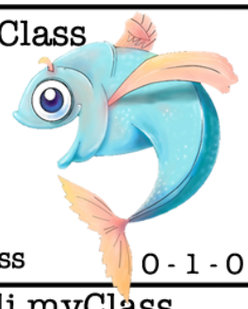
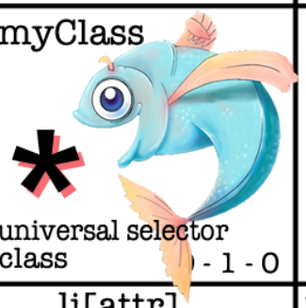
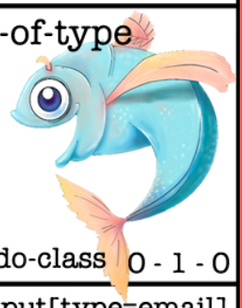
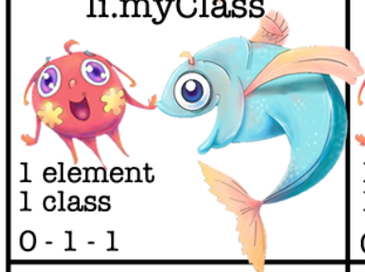
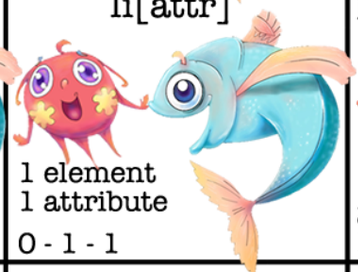
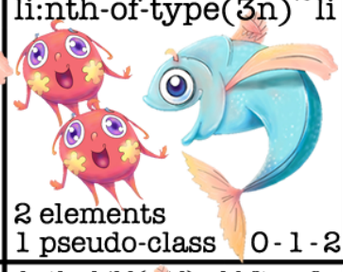
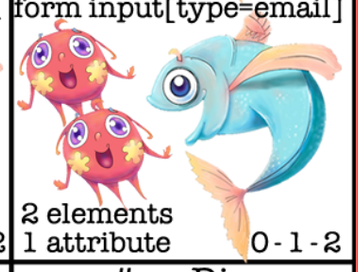
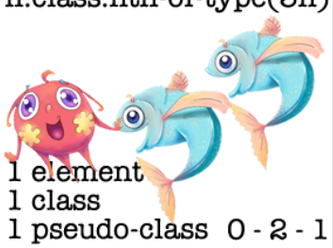
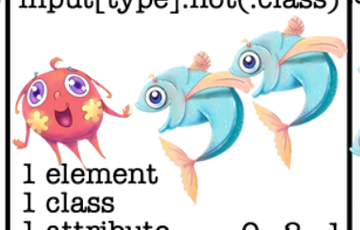
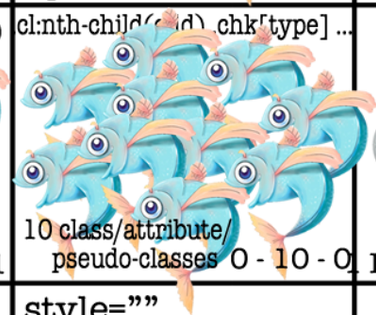
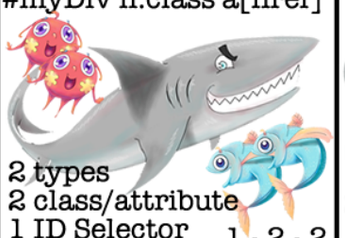
Which of these wins?

```
.green { color: green; }
.red { color: red; }
```

`.red` wins here, it doesn't matter the order in the `class` attribute, all that matters is the order the styles are defined in the CSS itself.

CSS SpeciFISHity

WITH PLANKTON, FISH AND SHARKS

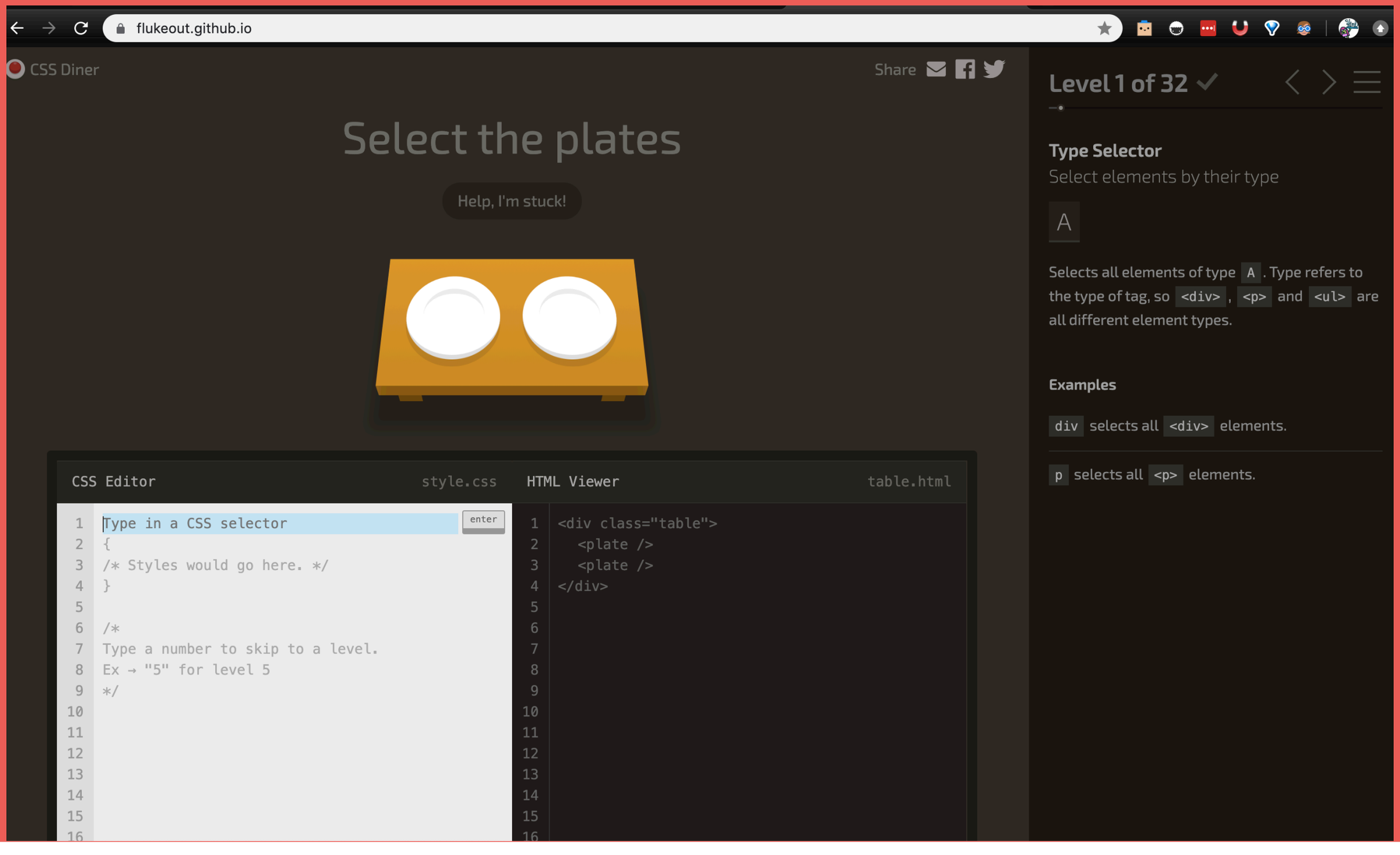
*  universal selector 0 - 0 - 0	div  1 element 0 - 0 - 1	li > ul  2 elements 0 - 0 - 2	body div ... ul li p a  12 elements 0 - 0 - 12
.myClass  1 class 0 - 1 - 0	*.myClass  1 universal selector 1 class 0 - 1 - 0	[type=checkbox]  1 attribute selector 0 - 1 - 0	:only-of-type  1 pseudo-class 0 - 1 - 0
li.myClass  1 element 1 class 0 - 1 - 1	li[attr]  1 element 1 attribute 0 - 1 - 1	li:nth-of-type(3n)~li  2 elements 1 pseudo-class 0 - 1 - 2	form input[type=email]  2 elements 1 attribute 0 - 1 - 2
li.class:nth-of-type(3n)  1 element 1 class 1 pseudo-class 0 - 2 - 1	input[type]:not(.class)  1 element 1 class 1 attribute 0 - 2 - 1	cl:nth-child(...id) chk[type] ...  10 class/attribute/ pseudo-classes 0 - 10 - 0	#myDiv  ID Selector 1 - 0 - 0
#myDiv li.class a[href]  2 types 2 class/attribute 1 ID Selector 1 - 2 - 2	#divitis #myDiv a  2 ID Selectors 1 type selector 2 - 0 - 1	style=""  inline style 1-0-0-0	!important  !important 1 - 0 - 0 - 0 - 0

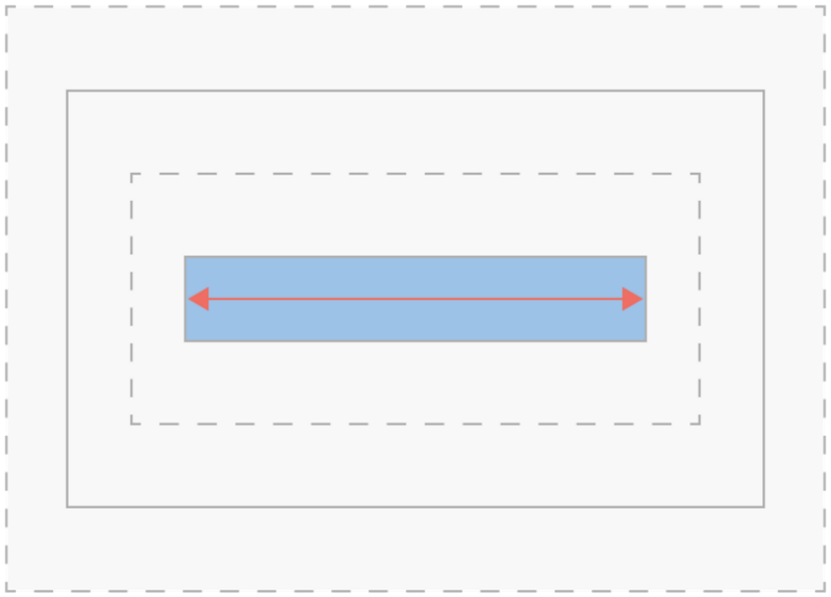
ESTELLE WEYL * @ESTELLEVW * WWW.STANDARDISTA.COM * 2104

X-0-0: The number of ID selectors
0-Y-0: The number of class selectors, attributes selectors, and pseudo-classes
0-0-Z: The number of type selectors and pseudo-elements
*, +, >, ~ : Universal selector and combinators do not increase specificity
:not(x): Negation selector has no value. Argument increases specificity

CSS

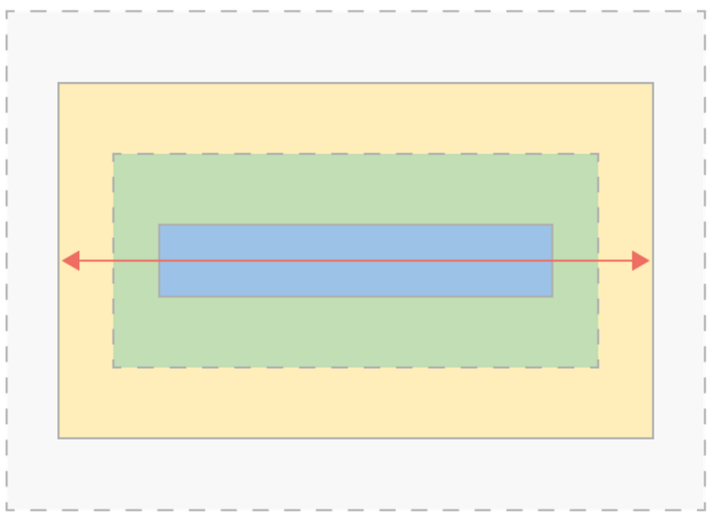
Advanced Specificity Challenge





Without box-sizing: border-box

Margin, borders and padding are drawn outside the set width of your content.



With box-sizing: border-box

Borders and padding are drawn inside the set width of your content. The margin is drawn outside.

CSS

Advanced Topic

Use box-sizing border-box

You can choose whether or or not to include borders and padding in the width of your content.

```
*, *:before,
*:after {
-moz-box-
sizing: border-
box;
-webkit-box-
sizing:
border=box
box-sizing:
border-box;
}
```

Place at the top of your CSS file. The * will target all elements on the page.at the top of your CSS file. The * will target all elements on the page.