Web Skills Day 2

SVG

SVG

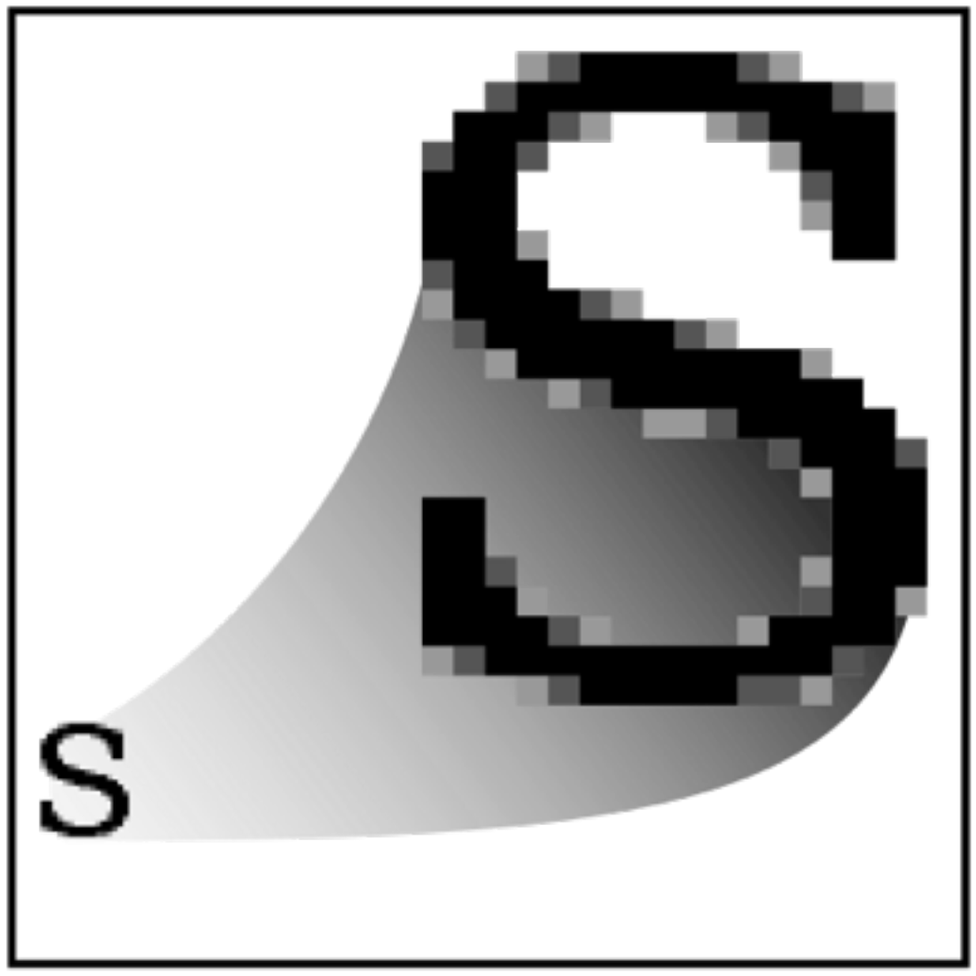
SVG stands for Scalable Vector Graphics

SVG is used to define vector-based graphics for the Web

SVG defines the graphics in XML format

SVG graphics do NOT lose any quality if they are zoomed or resized

Every element and every attribute in SVG files can be animated



Raster

.jpeg .gif .png

Vector

.svg

SVG: How To Use

```
<!DOCTYPE html>
<html>
<body>

<h1>My first SVG</h1>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>

</body>
</html>
```

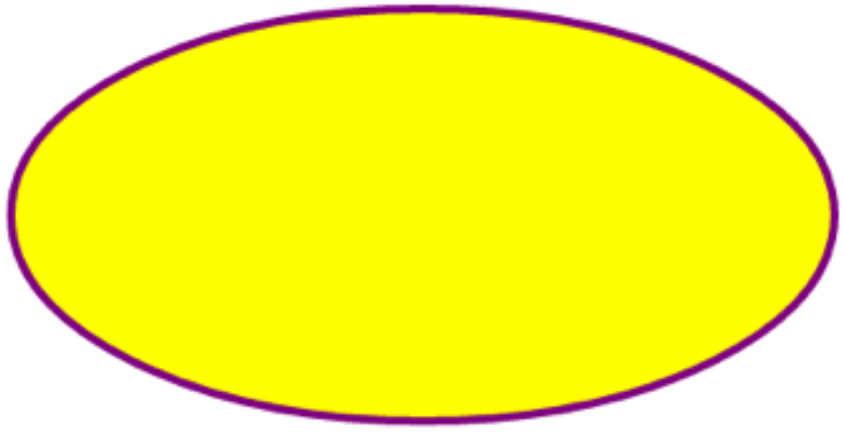
My first SVG

SVG: Text

```
<!DOCTYPE html>
<html>
<body>

<svg height="30" width="200">
  <text x="0" y="15" fill="red">I love SVG!</text>
</svg>

</body>
</html>
```

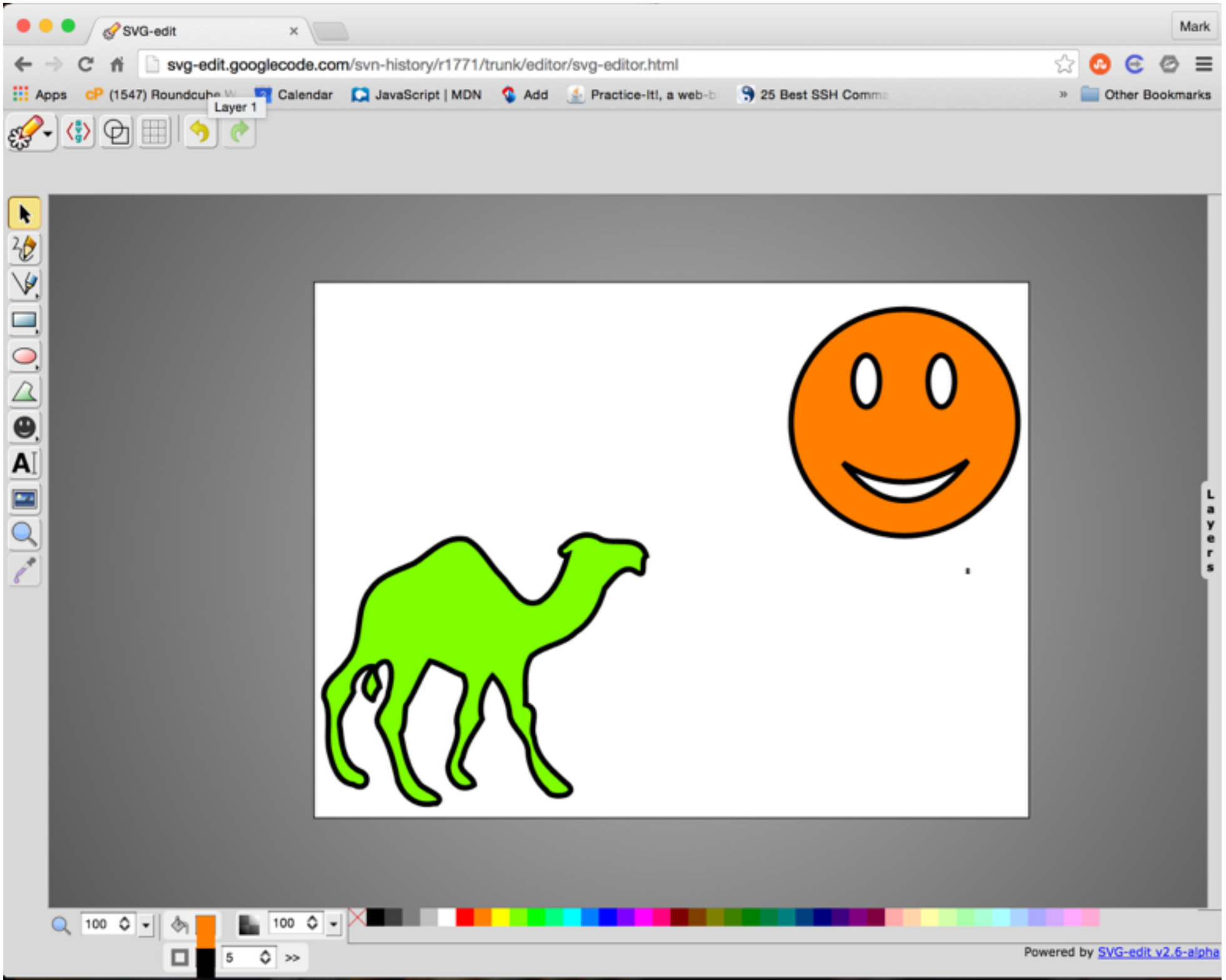
SVG: Circle

```
<!DOCTYPE html>
<html>
<body>

<svg height="140" width="500">
  <ellipse cx="200" cy="80" rx="100" ry="50"
style="fill:yellow;stroke:purple;stroke-width:2" />
</svg>

</body>
</html>
```

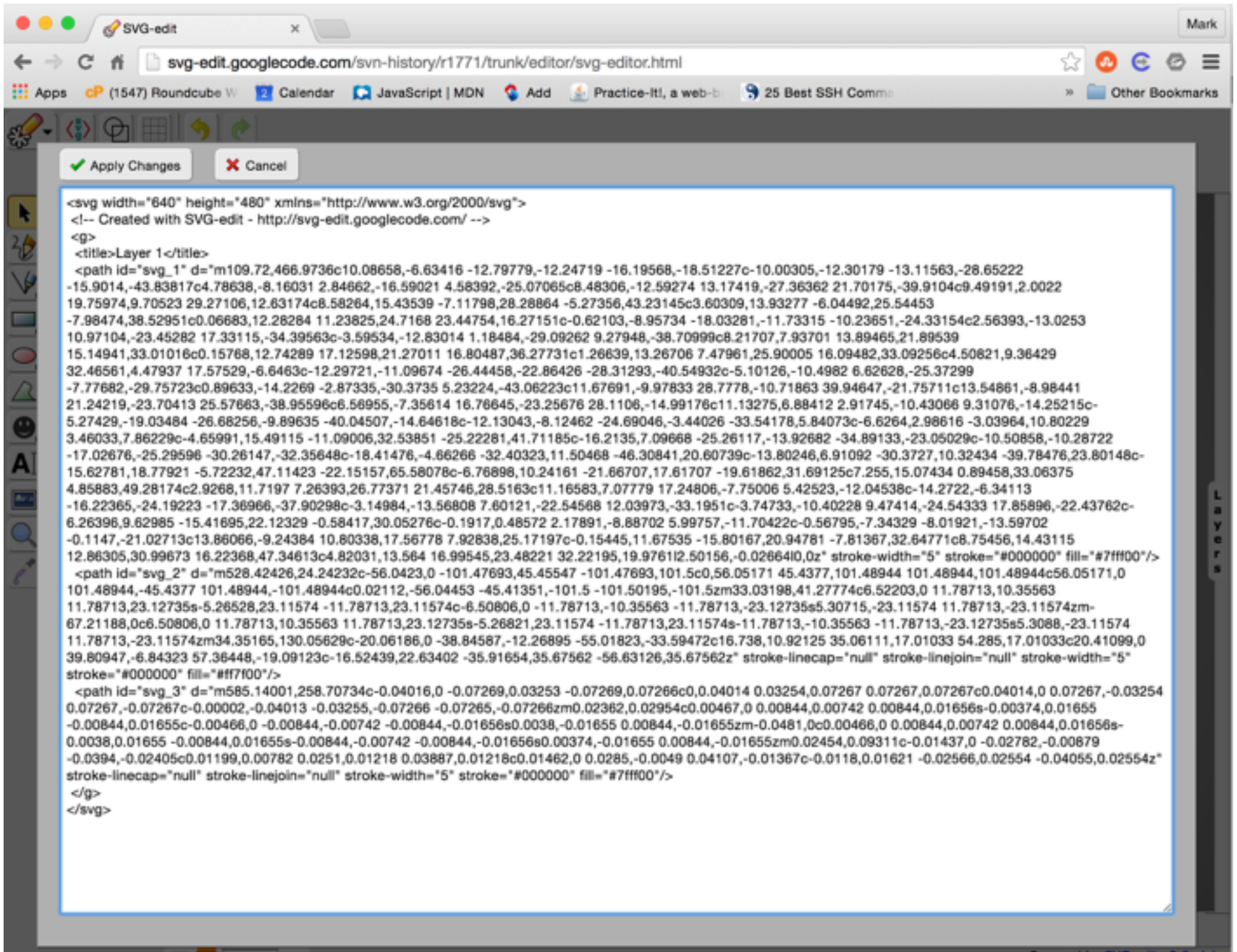


Online SVG Editor:

http://svg-edit.googlecode.com/svn-history/r1771/trunk/editor/svg-editor.html

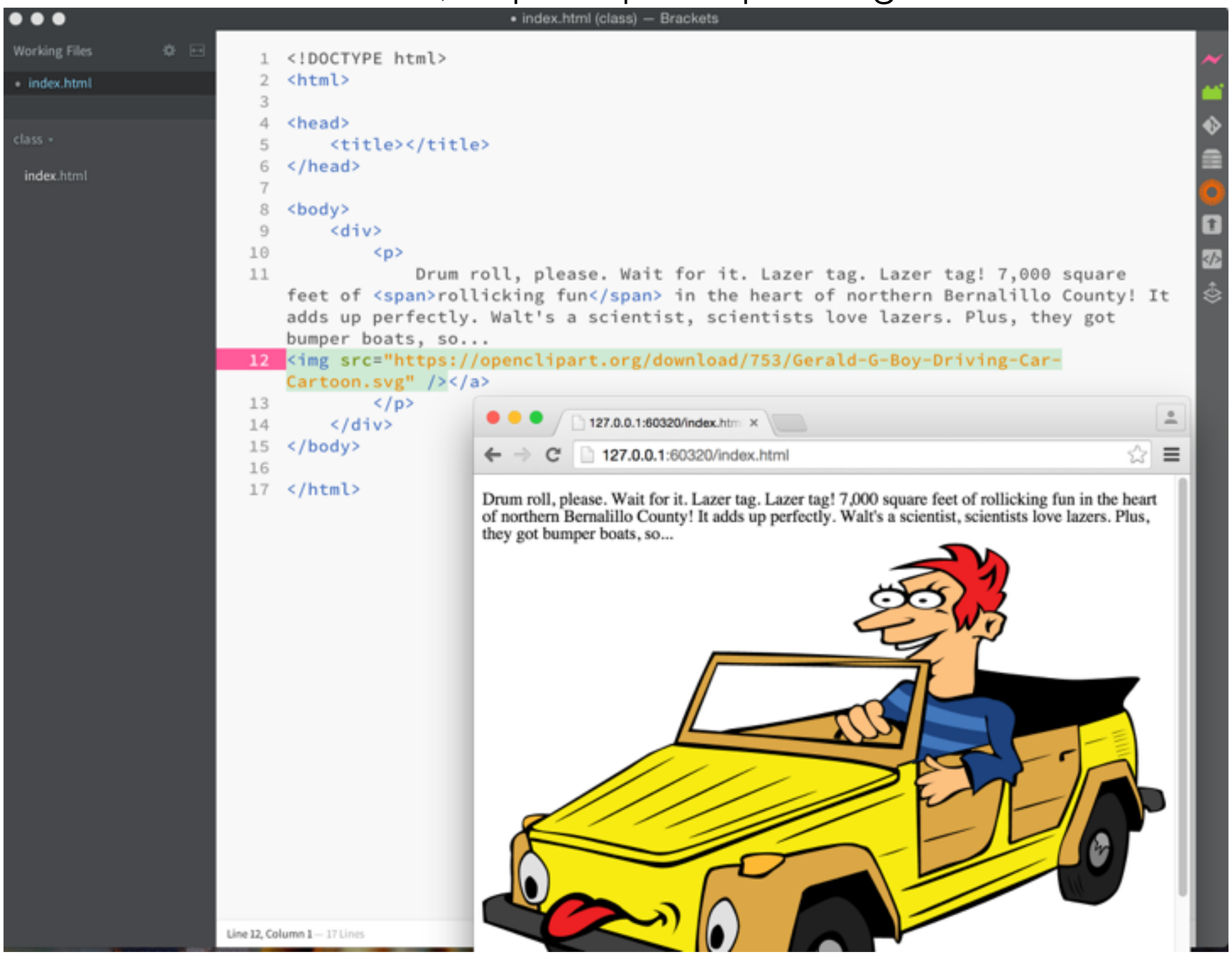


Online SVG Editor: http://svg-edit.googlecode.com/svn-history/r1771/trunk/editor/svg-editor.html



SVG

you can also embed svg files in an image tag
SVG files can be exported from popular programs like Illustrator
Also, https://openclipart.org/



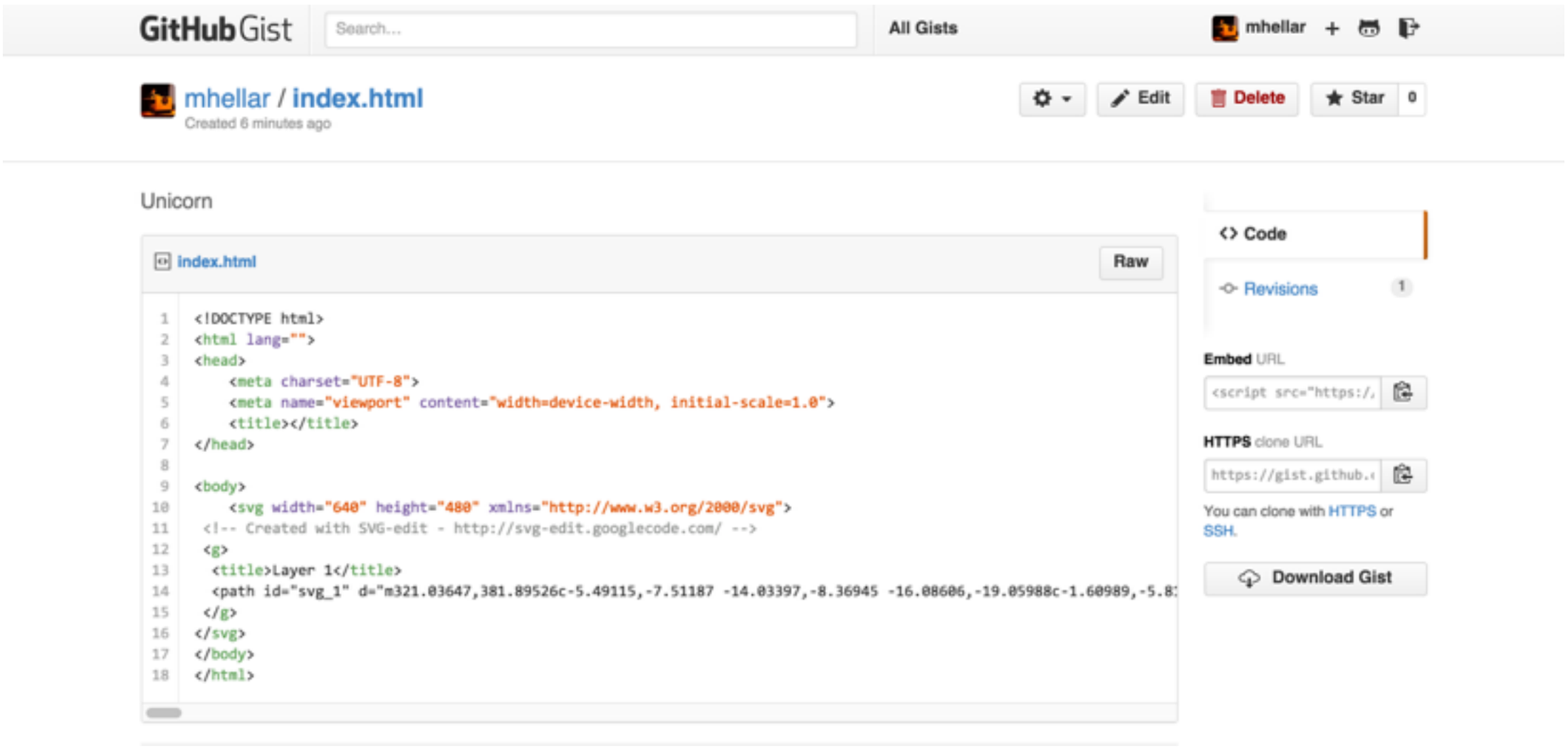
Sharing Your Work with Gists and Blocks

mhellar's block #e5bfecaf53471bb69fdf April 7, 2015

Unicorn

Open in a new window.

Sharing Your Work with Gists and Blocks



1 Go to https://gist.github.com/

2 Enter a description, name the file index.html and paste the code you want to share

Sharing Your Work with Gists and Blocks

You will get a URL like this:

https://gist.github.com/mhellar/e5bfecaf53471bb69fdf

Replace gist.github.com with this bl.ocks.org

http://bl.ocks.org/mhellar/e5bfecaf53471bb69fdf

Tada!!!!!

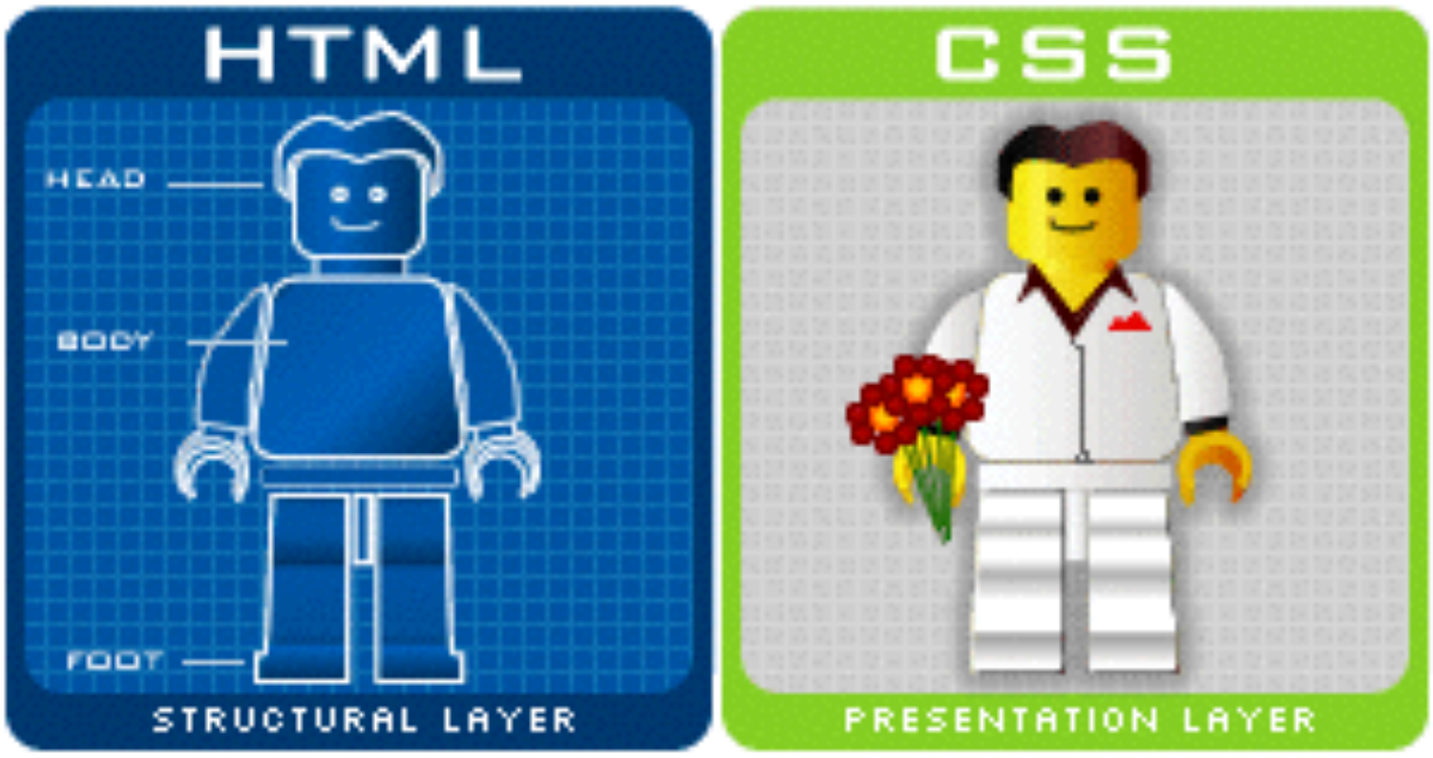
mhellar's block #e5bfecaf53471bb69fdf April 7, 2015

Unicorn

Open in a new window.

Assignment, craft a SVG composition and share it with Bl.ock to the group..

Cascading Style Sheets(CSS)



Cascading Style Sheets(CSS)

It is best to think of web design as three layers of disciplines:

1. Content and Structure : HTML

2. Visual Style and Presentation : CSS

3. Interactivity, Motion and Function : Javascript(next week)

Cascading Style Sheets(CSS)

You can say that **HTML** is like building the **structure** of the house, You need to create a skeleton of where everything is going to go.

CSS is the decorating step. Now you're concerned with the **visual style** and how to make things more presentable.

And finally all the other programming languages and "bells and whistles" such as **Javascript**, Flash, video, etc. are like adding the amenities. In web design this means adding **interactivity**—you add cool things like animations, feedback forms, and other useful functions that make your website more complete.

Cascading Style Sheets(CSS)

CSS can define these general attributes of HTML tags:

Typography & Lists

- font, font size, line spacing, etc.
- bullets, spacing

Colors & Ornaments

- color, background color, border color, etc.
- background images

Positioning & Layout

- block size, width, height
- margin, padding, absolute positioning

Miscellaneous (used or special attributes)

- printing instructions, pseudo classes

A few new HTML Tags:
and

CSS and other web-centric programing languages have prompted the creation of two new “anonymous” tags: div and span. These tags when used on their own don’t modify the content, but when used in conjunction with CSS, can be used as “custom” tags. div is a block level tag, while span is an inline tag.

```
<!DOCTYPE html>
<html>

<head>
    <title></title>
</head>

<body>
    <div>
        <p>
            Drum roll, please. Wait for it. Lazer tag. Lazer tag! 7,000 square
feet of <span>rollicking fun</span> in the heart of northern Bernalillo County! It
adds up perfectly. Walt's a scientist, scientists love lazers. Plus, they got
bumper boats, so...

        </p>
    </div>
</body>

</html>
```

A few new HTML Tags:
and

div

As stated earlier, use the `div` tag to define a “chunk” of information. `div`s are great for delineating sections of your web site.

span

Use `span` to mark up words or certain portions of text inside of a block level element.

```
<!DOCTYPE html>
<html>

<head>
    <title></title>
</head>

<body>
    <div>
        <p>
            Drum roll, please. Wait for it. Lazer tag. Lazer tag! 7,000 square
feet of <span>rollicking fun</span> in the heart of northern Bernalillo County! It
adds up perfectly. Walt's a scientist, scientists love lazers. Plus, they got
bumper boats, so...

        </p>
    </div>
</body>

</html>
```

Anatomy of a CSS Definition

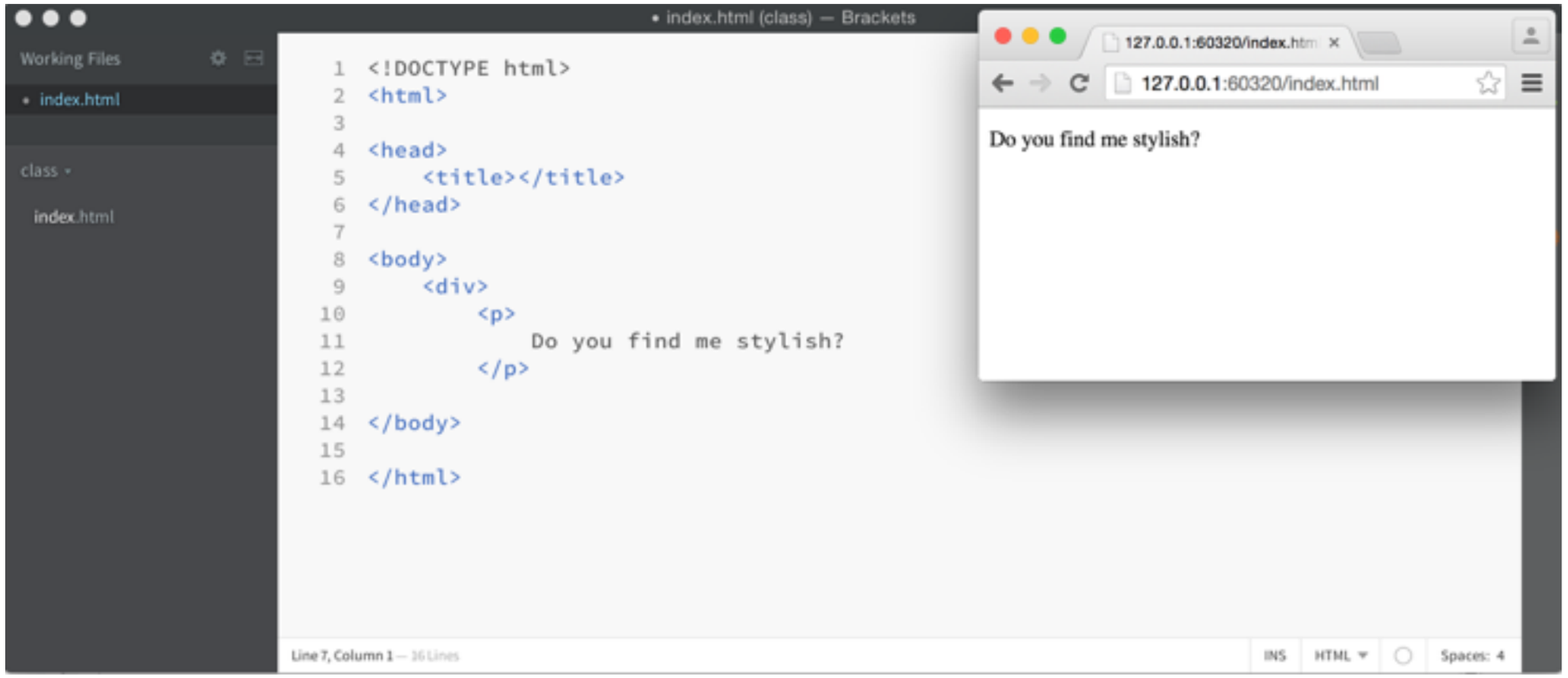
```
selector {
property: value; <- This line is called a rule.
}
```

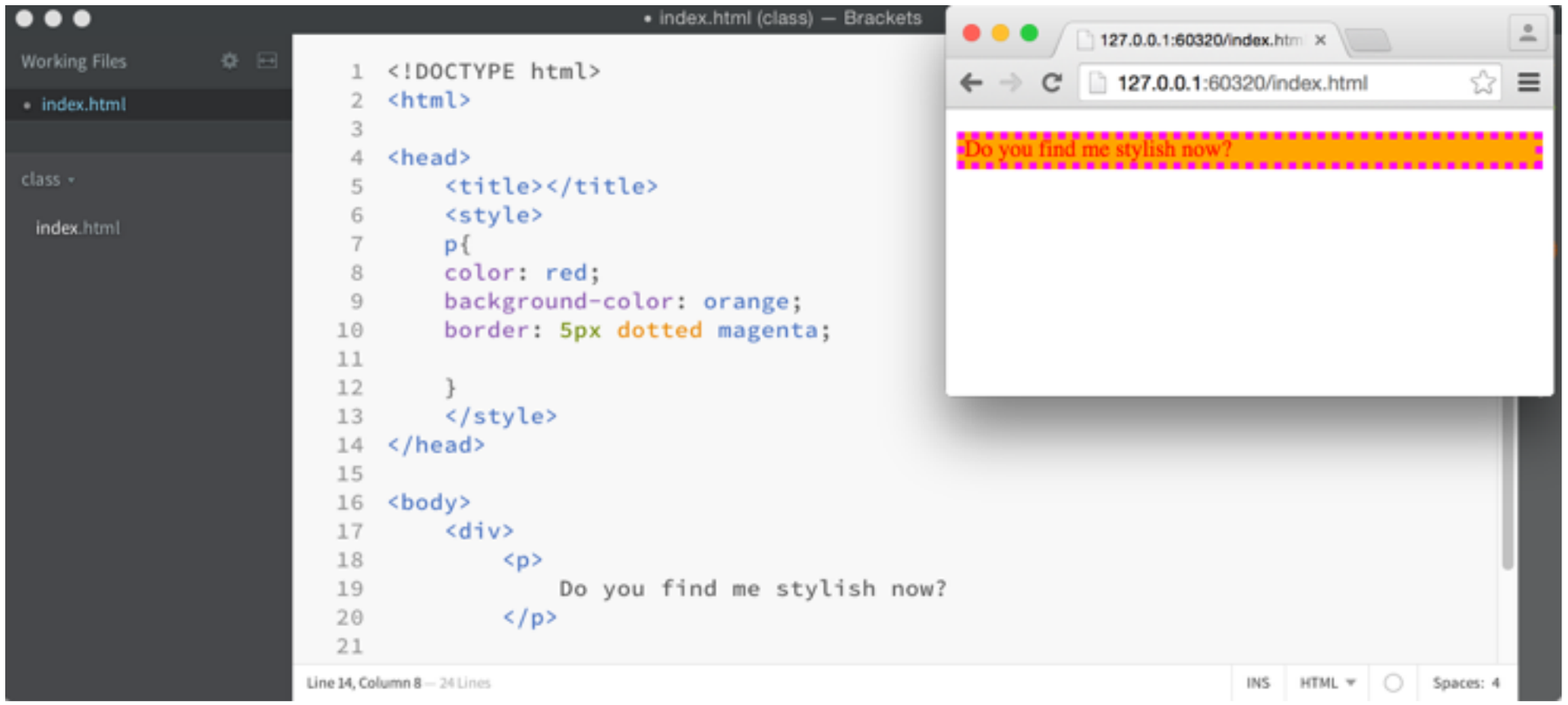
This code isn't valid CSS, but the code in red defines the name of the part of the code.

Cascading Style Sheets(CSS)

```
/*
      CSS Code Example
*/
body {
      font-family: Verdana, Arial, Helvetica, sans-serif;
      font-size: 76%;
      margin: 0 0 0 0;
      padding: 0 0 0 0;
}
h1 {
      font-family: Georgia, ‘Times New Roman’, Palatino, serif;
      font-size: 18px;
      border-bottom: 1px solid #333;
}
```

Cascading Style Sheets(CSS)





Where does CSS live?

CSS is read by web browsers in the following order:

-CSS code added in the HTML document head:

```
<head>
    <style type="text/css">
        h1 { font-size: 24px; }
    </style>
</head>
```

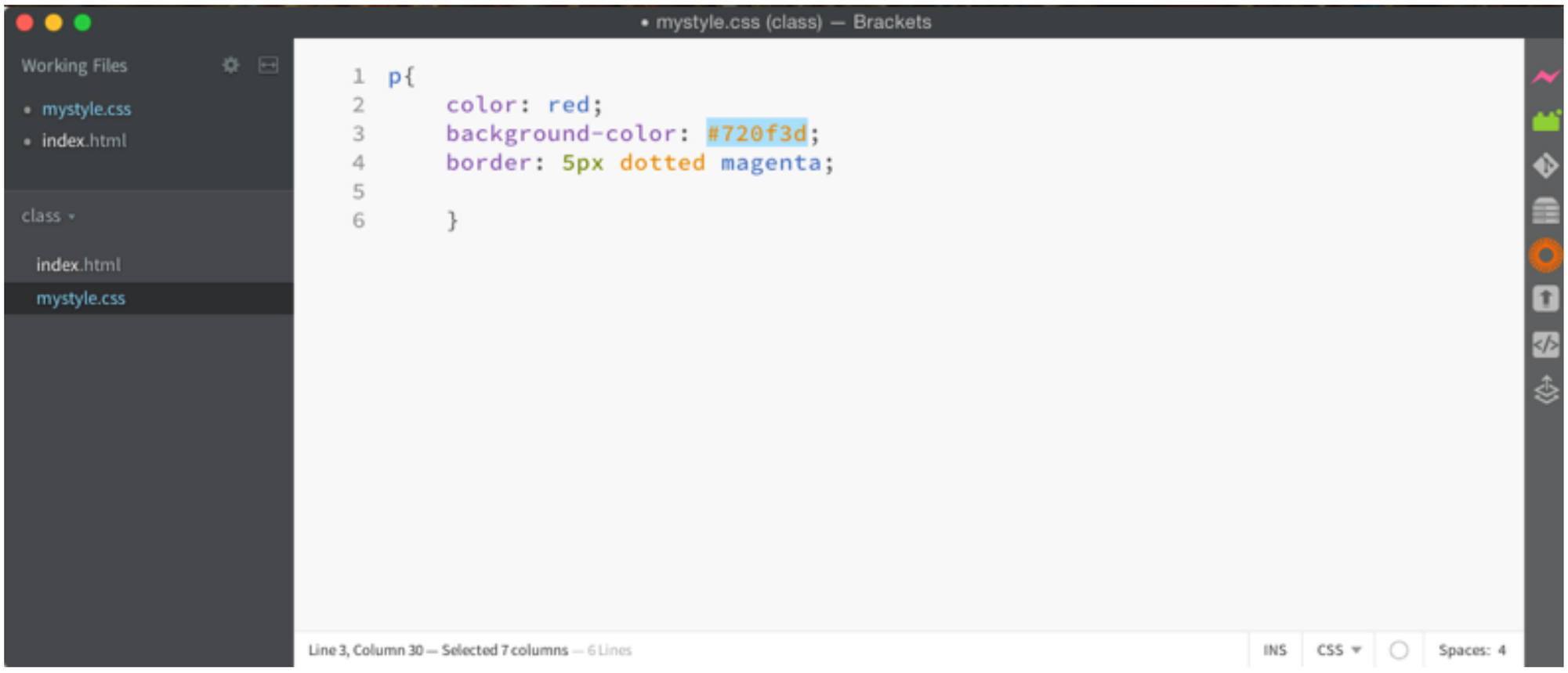
-external style sheet definitions (optimal)

```
<head>
    <link href="stylesheet.css" rel="stylesheet" type=“text/css">
</head>
```

-inline styles

```
<body>
  <h1 style=“font-size: 24px”>heading</h1>
</body>
```

Where does CSS live?



Naming your elements:

Naming your elements

Any tag in HTML can be given a name or “identifier.” This is useful in defining custom tags or pointing to a specific default element. There are two ways to name or identify an element:

- id
- class

```
<div id=”section”>
      <p class=”special”>Hi!!</p>
</div>
```

In this example, `div` *is given an* `id` *of “section”, while* `p` *is given a* `class` *of “special.” Think of these as just names.*

Naming your elements:

Naming tips

- `id` and `class` names can contain letters, numbers and dashes **only**
- names are case-sensitive
- names **cannot** start with a number or dash
- pick names that specify the function of the tag, not the visual presentation (i.e., “important” is a better name than “red”)

Difference between id and class

Quick explanation: Use `id` to name an element that is used **only once** on a web page. Use `class` to name an element that may be used **more than once** on a web page.

Referencing ids and classes in CSS

- use # (number sign) to identify an id
- use . (dot) to identify a class

Naming your elements:

CSS code example

(referring to last HTML example)

```
<style>
div#section {
	border: 1px solid #ddd;
	padding: 12px;
	margin-left: 32px;
}
p.special {
	font-size: 48px;
color: red; }
</style>

<div id=”section”>
	<p class=”special”>Hi There!!</p>
</div>
```

CSS Reference:

http://www.w3schools.com/cssref/

http://www.smashingmagazine.com/wp-content/uploads/images/css3-cheat-sheet/css3-cheat-sheet.pdf

Cascading Style Sheets(CSS)

CSS Rules: Typography & Lists

Typography

font-family

```
{ font-family: Verdana, ‘Trebuchet MS’, Arial, sans-serif; }
```

font-size

```
{ font-size: 18px; }
```

letter spacing / word spacing

```
{ letter-spacing: 2px; word-spacing: 14px; }
```

line-height / text transform

```
{ line-spacing: 5em; text-transform: uppercase; }
```

Lists

list-style-type

```
{ list-style-type: disc; }
```

margin / padding

```
{ margin; 0; padding: 0; }
```

text-indent

```
{ text-indent; 12px; }
```

Cascading Style Sheets(CSS)

CSS Rules: Colors & Ornaments

Colors

color

```
{ color: #e0e }
```

background-color

```
{ background-color: #f0e; }
```

background-image

```
{ background-image: url(‘images/image.jpg’); }
```

Ornaments

border-size

```
{ border: 1px; }
```

border-style

```
{ border-style: dotted; }
```

border-color

```
{ border-color: red; }
```

border

```
{ border: 1px solid #ddd; }
```

CSS Rules: Miscellaneous

cursor

```
{ cursor: crosshair; }
```

text-shadow

```
{ text-shadow: black, 3px 3px ; }
```

visibility

```
{ visibility: none; }
```

Advanced Selectors

comma

```
h1, h2, h3 { color: red; }
```

spaces

```
div#main h1 { color: blue; }
```

pseudo-selectors

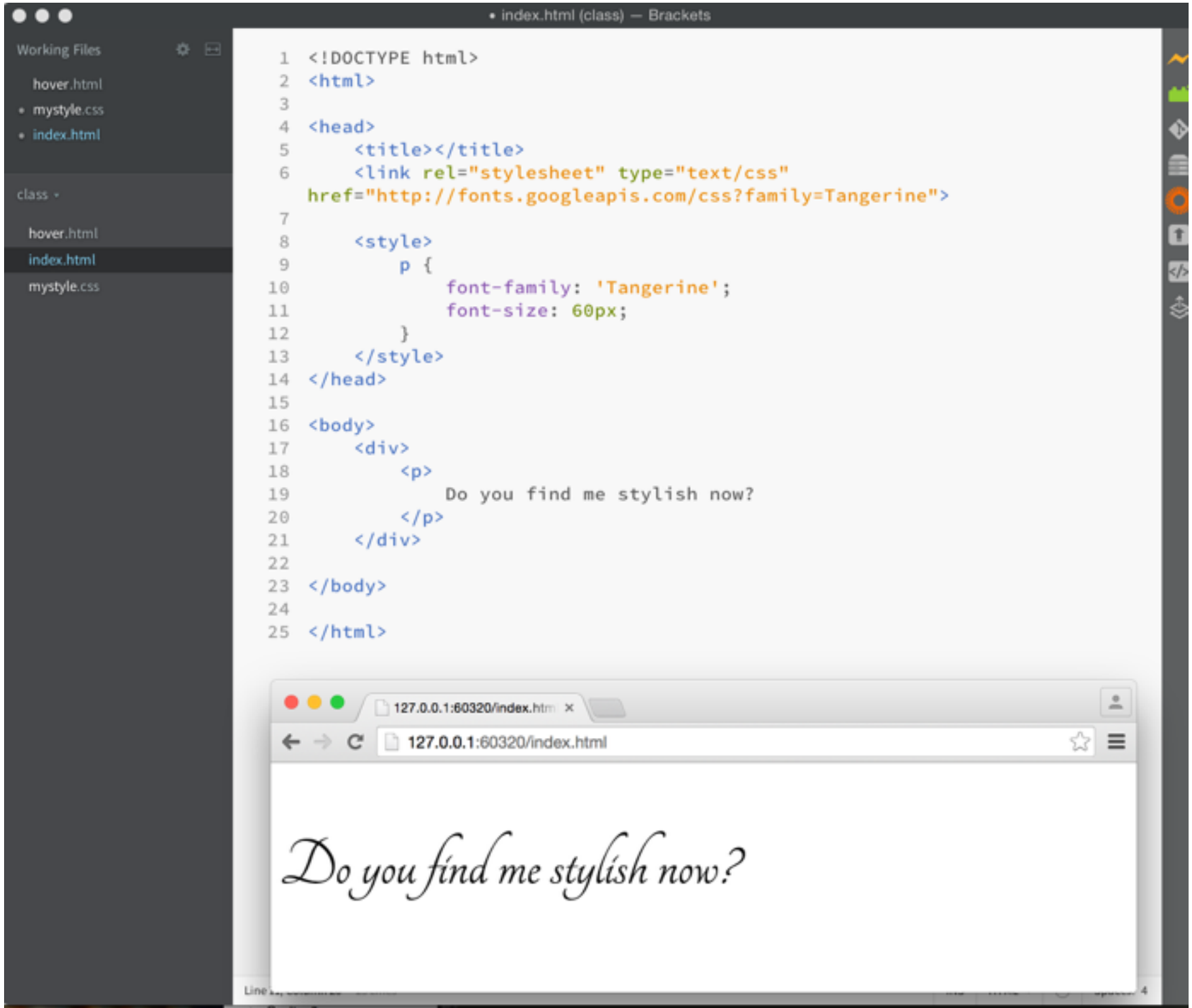
```
a:hover { color: #ddd; }
```

http://www.cheetyr.com/css-selectors

Web Fonts

http://www.google.com/fonts

http://hellohappy.org/beautiful-web-type/



CSS Animations:

http://www.w3schools.com/css/css3_animations.asp

```
/* The animation code */
@keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

CSS Animations:

http://www.w3schools.com/css/css3_animations.asp

```
 /* The animation code */
@keyframes example {
    0%   {background-color: red; left:0px; top:0px;}
    25%  {background-color: yellow; left:200px; top:0px;}
    50%  {background-color: blue; left:200px; top:200px;}
    75%  {background-color: green; left:0px; top:200px;}
    100% {background-color: red; left:0px; top:0px;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

Code Playgrounds

http://jsfiddle.net/

http://codepen.io/

http://cssdeck.com/

http://dabblet.com/

http://liveweave.com/

Cascading Style Sheets(CSS)

Syntax tips

- start with the element (tag) you will define, followed by opening and closing brackets
- within the brackets, end each definition with a semi-colon
- CSS is read from top to bottom — if you re-define an element twice, the last read definition will be applied, and the first will be ignored
- comments can be added by using /* */ tags — comments are helpful if collaborating with other designers and when looking at old code
- Be consistent with your coding habits