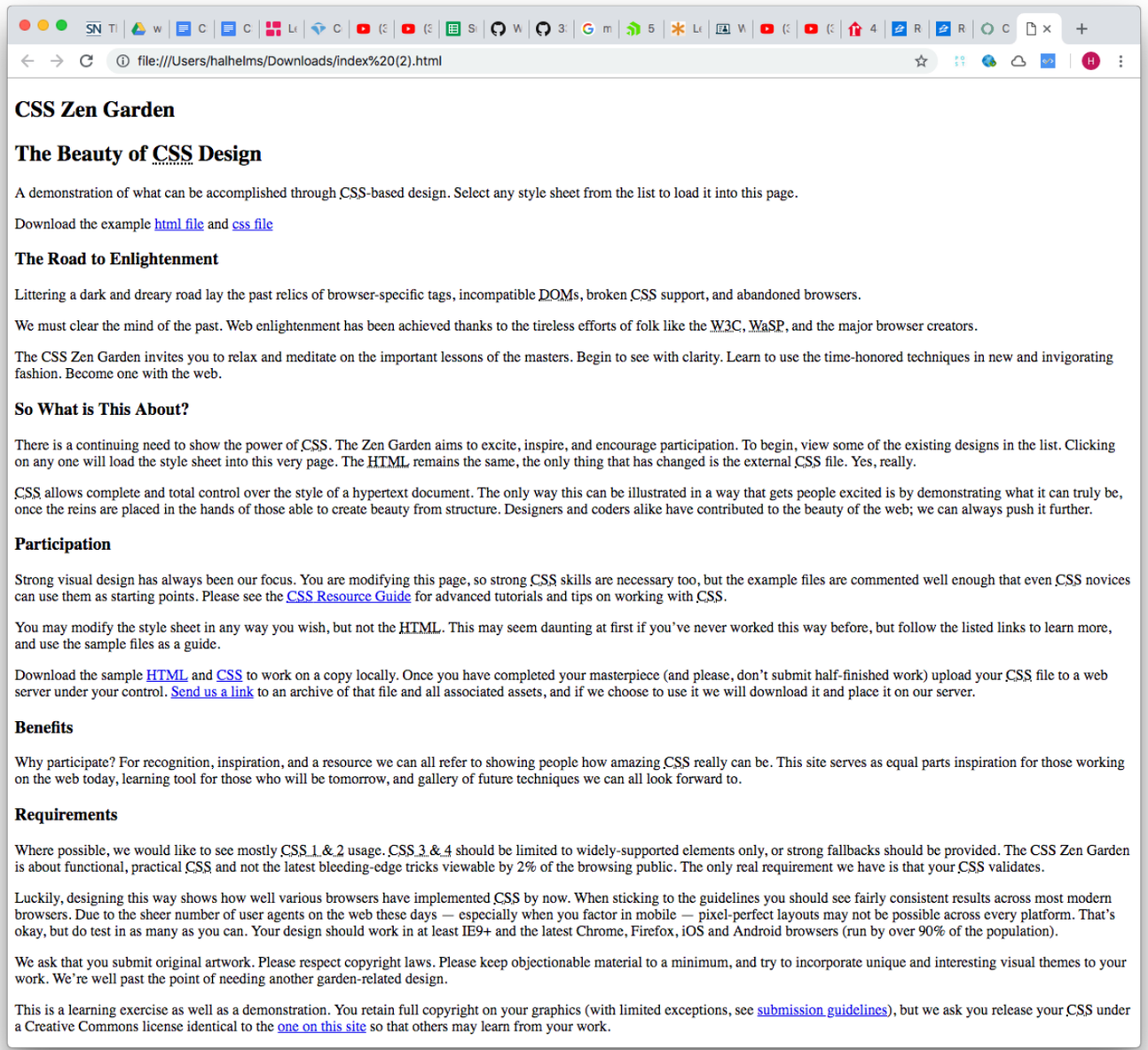
Intro to CSS

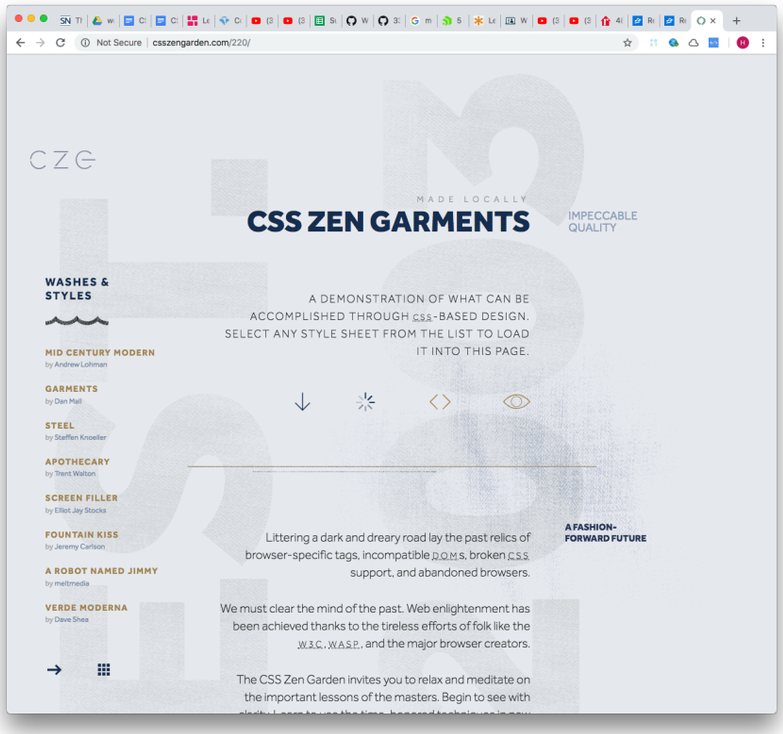
STYLING HTML

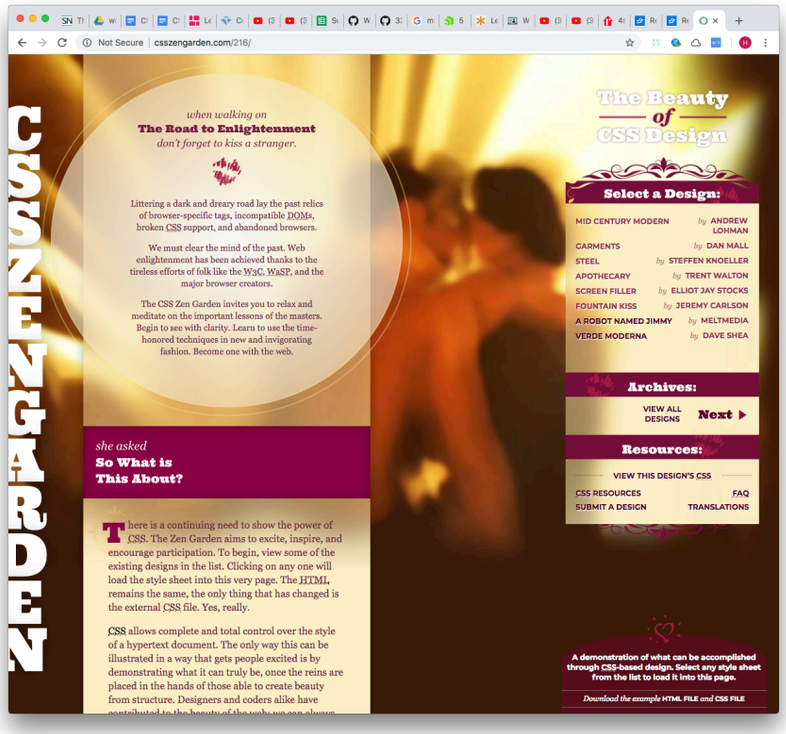
HTML might credibly be called "the language of the web". But, if we're honest, it's pretty boring stuff. It can't do much of anything without JavaScript. So, is it all good looks and no brains? Well — no, it's awfully *plain* as it is. In a minute, we're going to ask you to go to a website, but first, here's a sample of that site without any CSS styling:

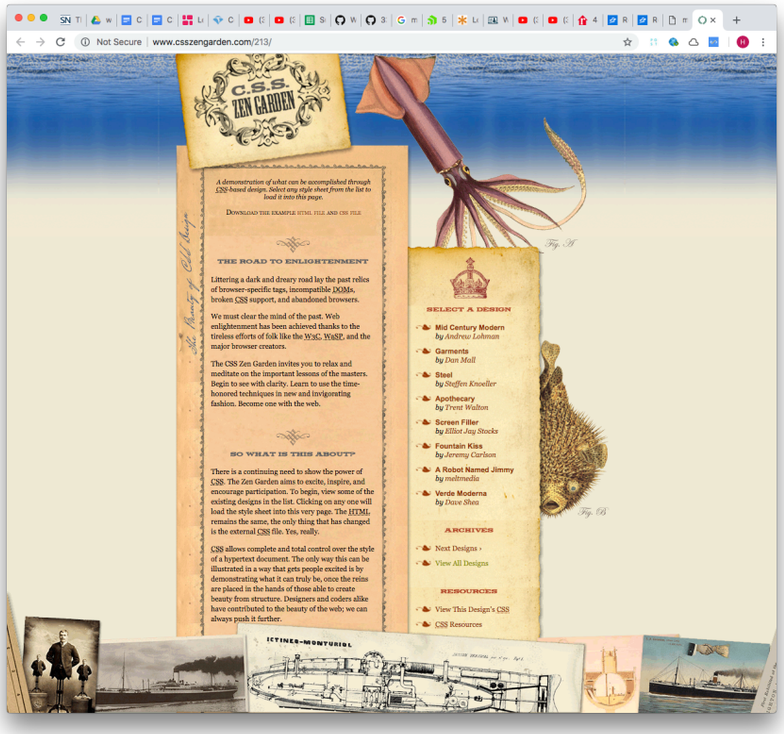


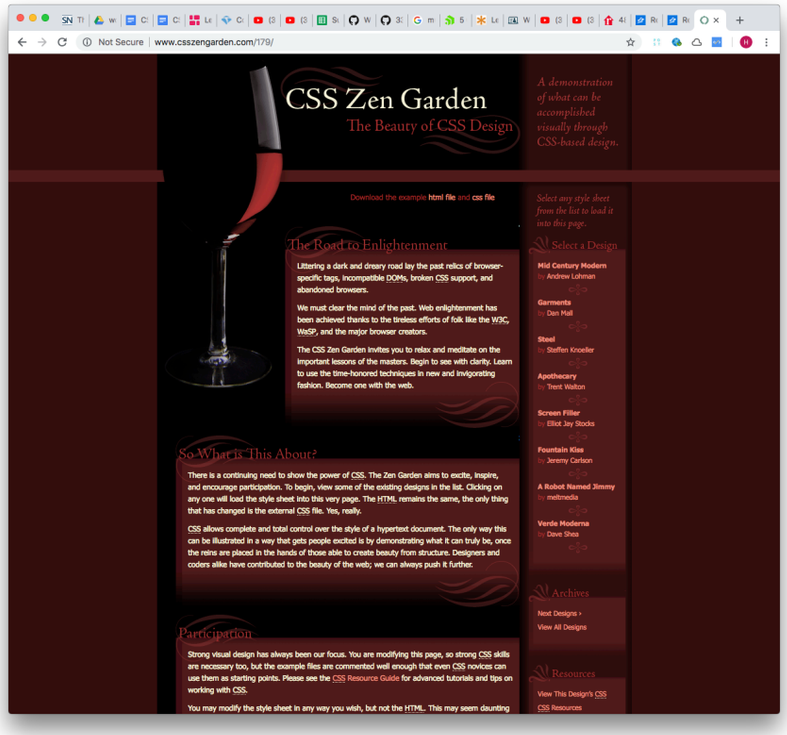
It provides structure (HTML is often compared to a human skeleton) but beauty? But here are snapshots of that very HTML — but with CSS added to it.

These four web pages all use exactly the same HTML. The only difference is the CSS applied to them.









These pages come from the site `csszengarden.com`. We urge you to go to the site, where you'll see over 200 different versions of that same page — all using the same markup, but different styling. That's the kind of impact CSS can have.

CSS RULESET

CSS styling is essentially simple: you *select* an element and then *style* the element by providing rules that guide its appearance. These are bundled together in a CSS *ruleset*. The following are various rulesets to give you a sense of how CSS is used. We'll look at specific selectors in subsequent sections of this Guide.

Selects all p elements

Sets text color to red

```
p {
color: red;
}
```

Selects the div element with an id of "main"

Sets font to "Arial"

```
div#main {
 font-family: "Arial";
}
```

Selects all li elements with a class of "active"

```
li.active {
 border: 1px solid grey;
}
```

Sets a 1 pixel wide solid grey border around the element

Selects all p tags that are children of div tags with a class of "special"

```
div.special>p {
 background-color: blue;
 color: white;
}
```

Makes the background color blue and the text color white

REFERENCE HTML

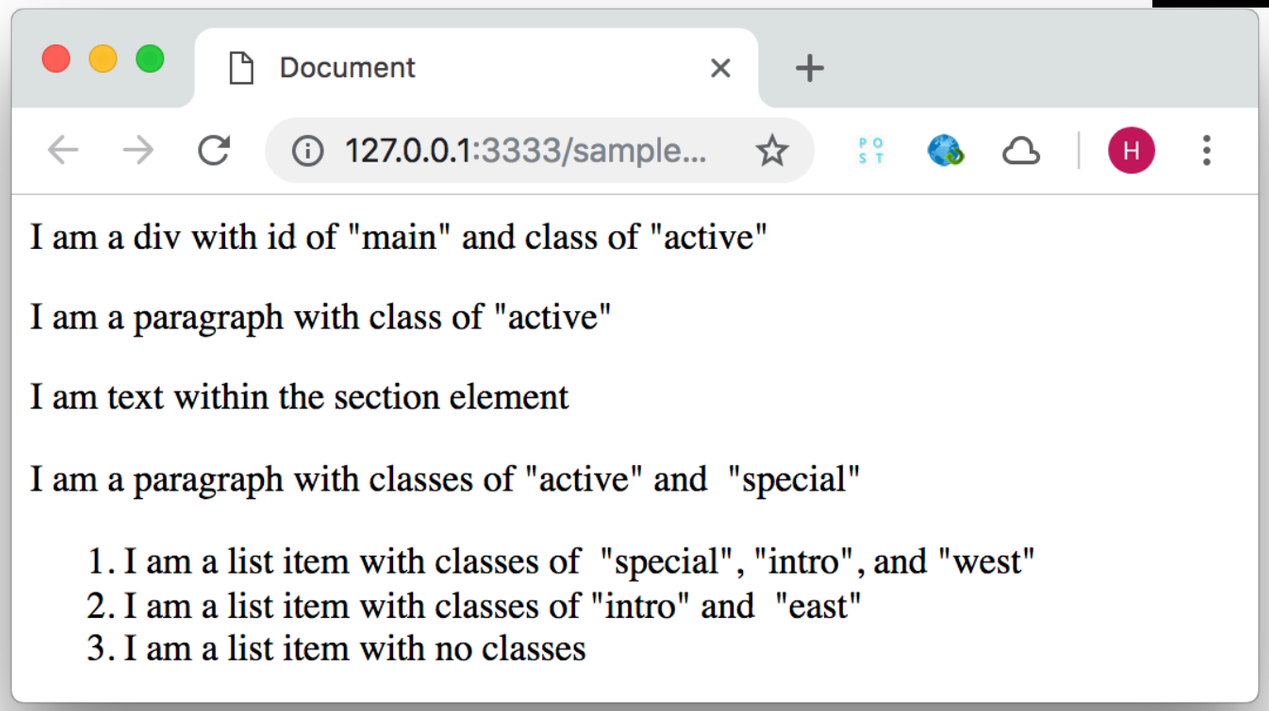
The following HTML will be used when working through various selectors.

Directories form a hierarchy of parents, children, grandchildren, etc. The "root" directory is the ultimate parent directory and, itself , has no parent.

```
<body>
 <div id="main" class="active">I am a div with id of "main" and class of
"active"</div>
 <p class="active">I am a paragraph with class of "active"</p>
 <section>
   I am text within the section element
   <p class="active special">I am a paragraph with classes of "active" and
 "special"</p>
   <ol>
     <li class="special intro west">I am a list item with classes of
 "special", "intro", and "west"</li>
     <li class="intro east">I am a list item with classes of "intro" and
 "east"</li>
     <li>I am a list item with no classes</li>
   </ol>
 </section>
</body>
```

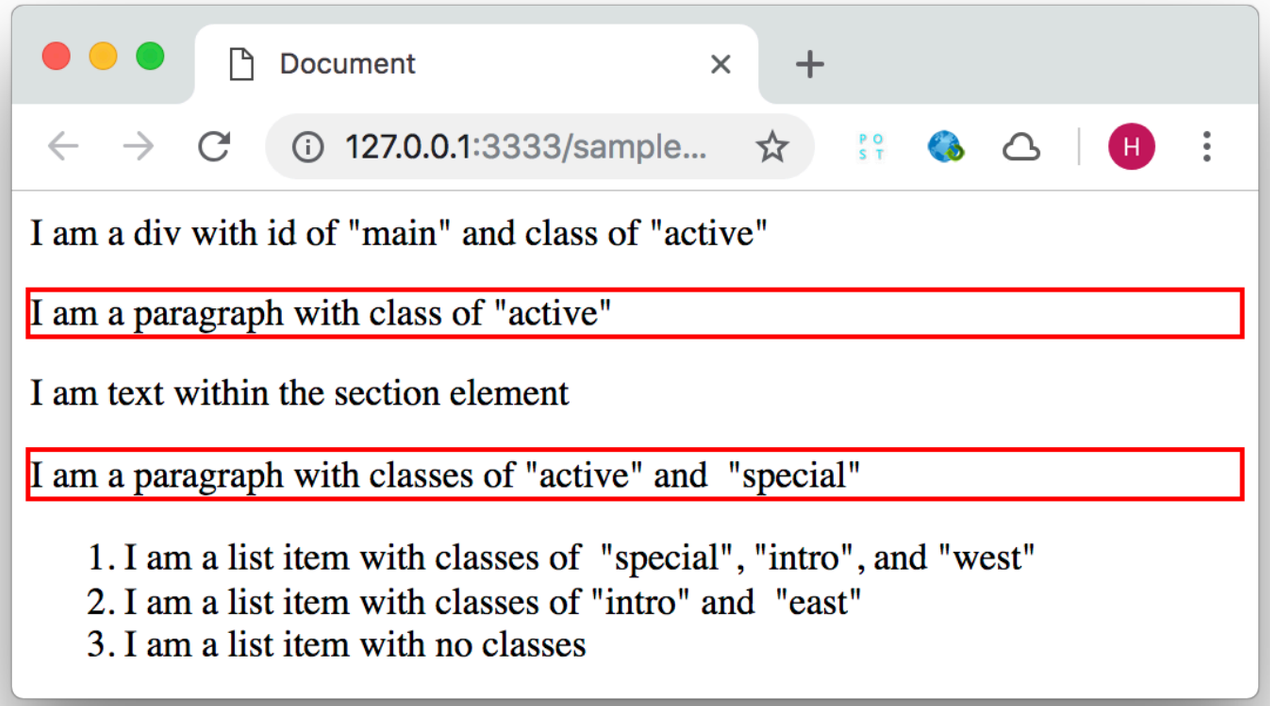
Although CSS is, as we said, simple, there are an enormous number of combinations of things you can do with it. Rather than trying to learn it all at once, we encourage you to learn things a little at a time, gradually acquiring mastery. In that spirit, our exploration of CSS will *not* attempt to be encyclopedic. We urge you to approach learning CSS in a spirit of *exploration*. When you see some effect you're baffled by, see how the designers performed their CSS magic.

Above HTML rendered by the browser



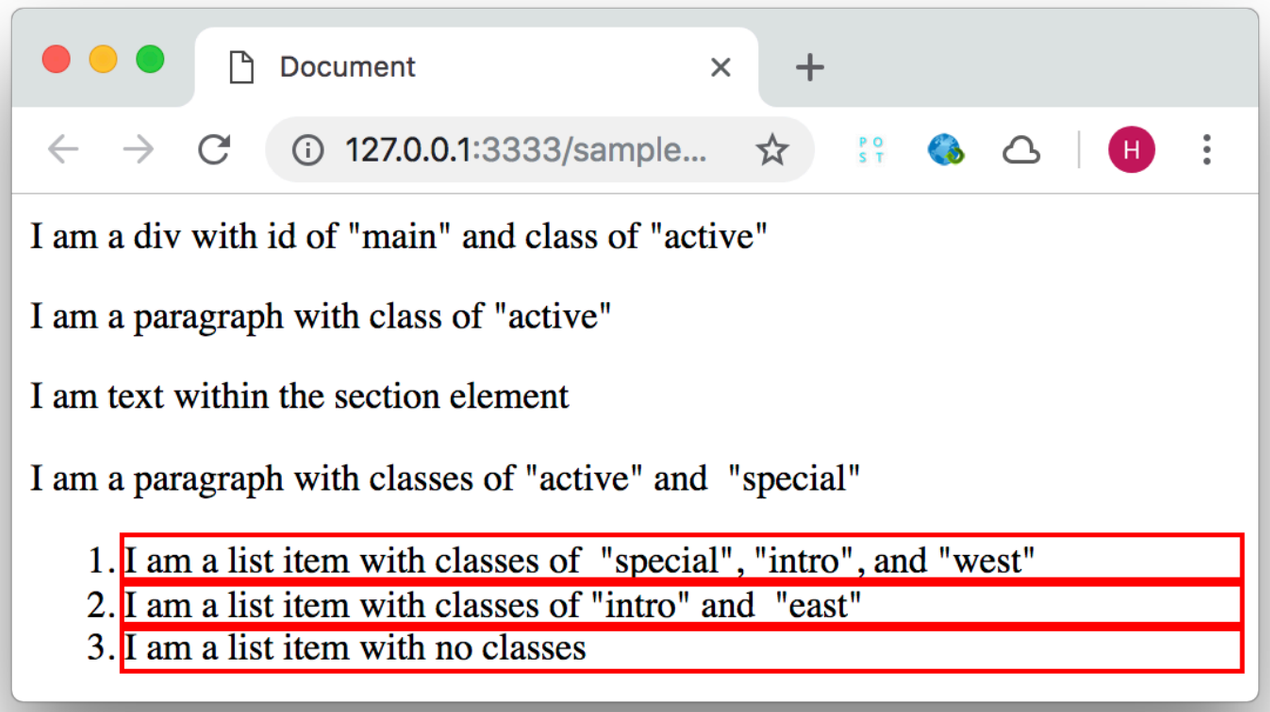
TYPE SELECTORS

The following pages show various type selectors. All are styled to have a 2 pixel solid red outline in order to make the selection clear.




```
p {
 outline: 2px solid red;
}
```

All <p> elements

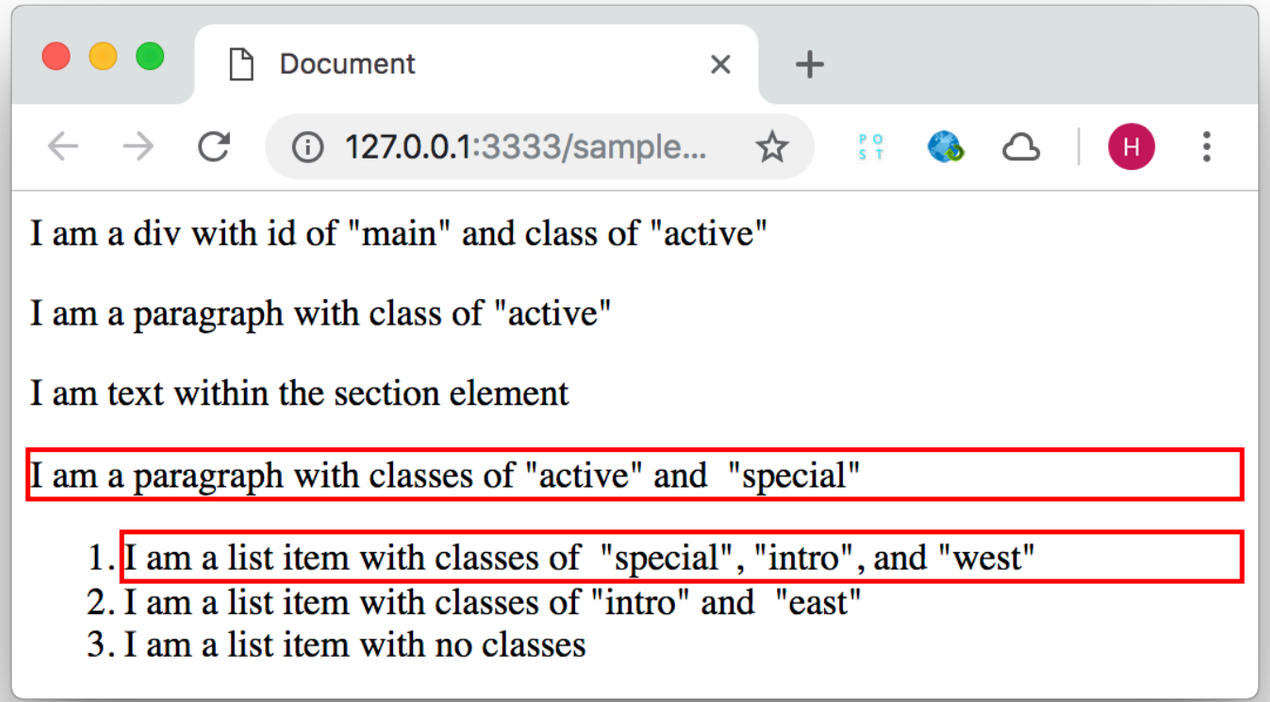



```
li {
 outline: 2px solid red;
}
```

All <li> elements

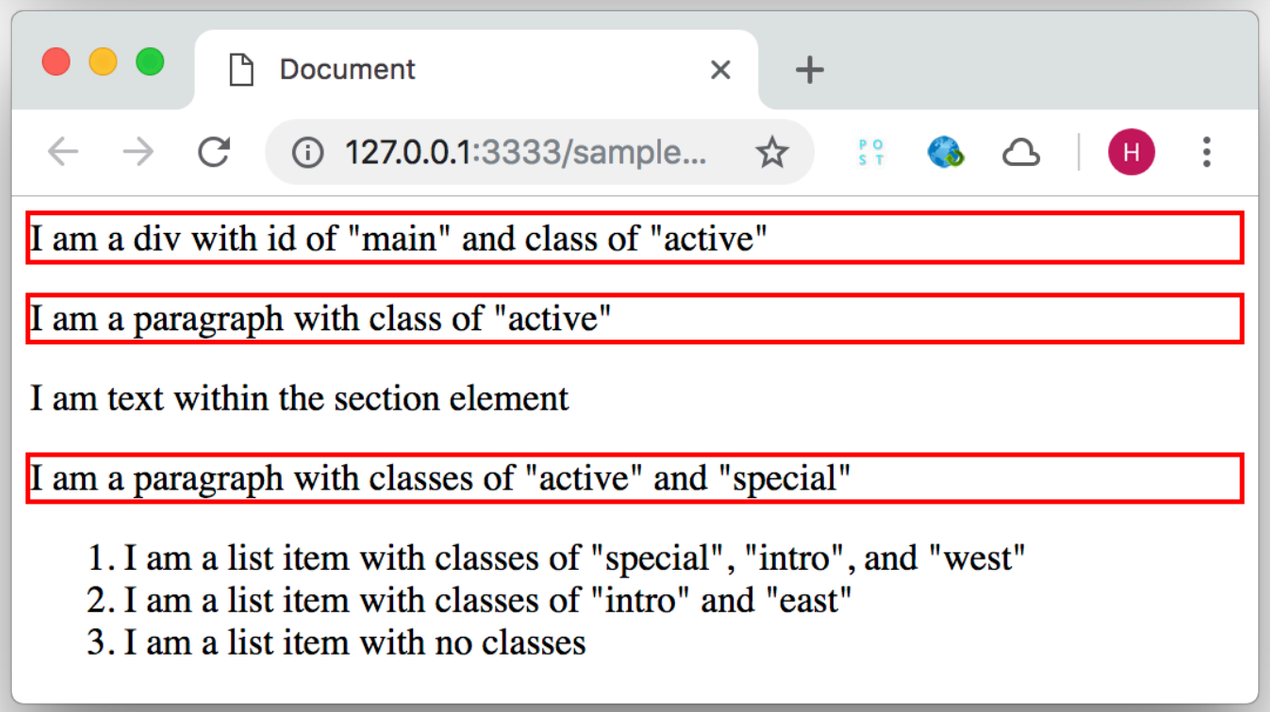
CLASS SELECTORS

The following shows various class selectors.




```
.special {
 outline: 2px solid red;
}
```

All elements of any type with a class of "special"




```
.active {
 outline: 2px solid red;
}
```

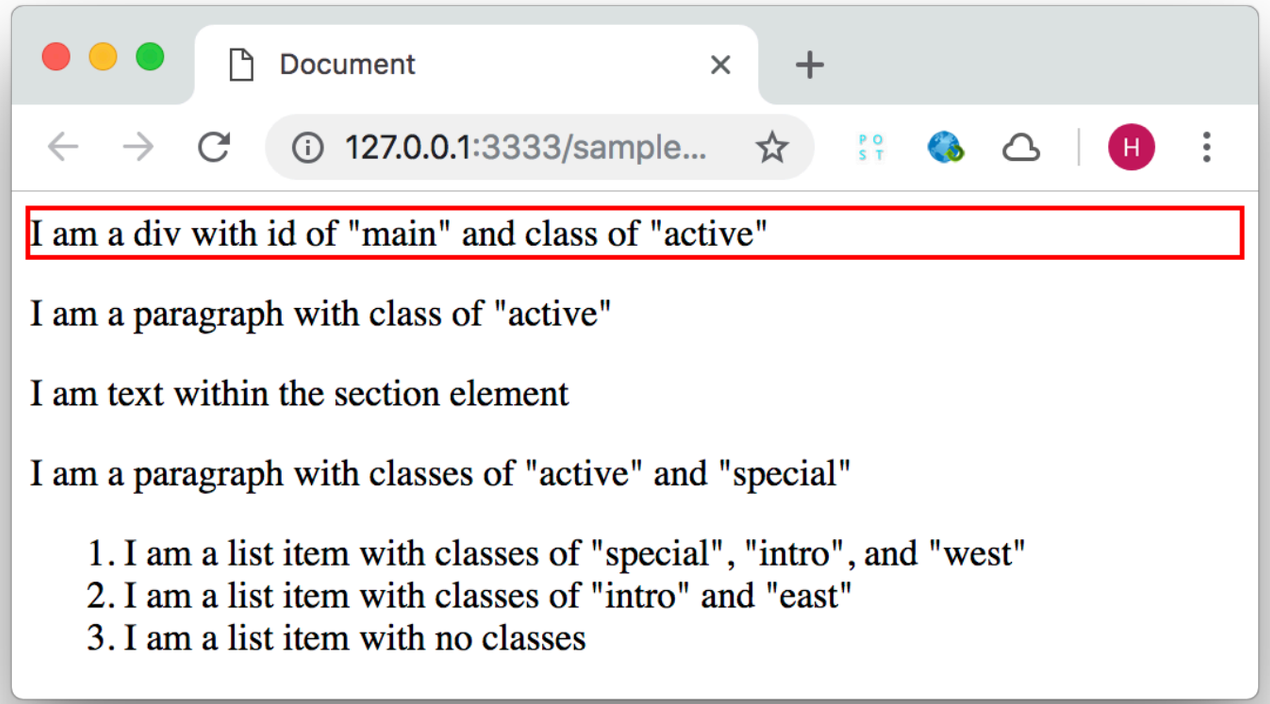
All elements of any type with a class of "active"

ID SELECTORS

The following shows various id selectors.

While many elements can have classes with the same value/s, all id values on the same page must be unique.

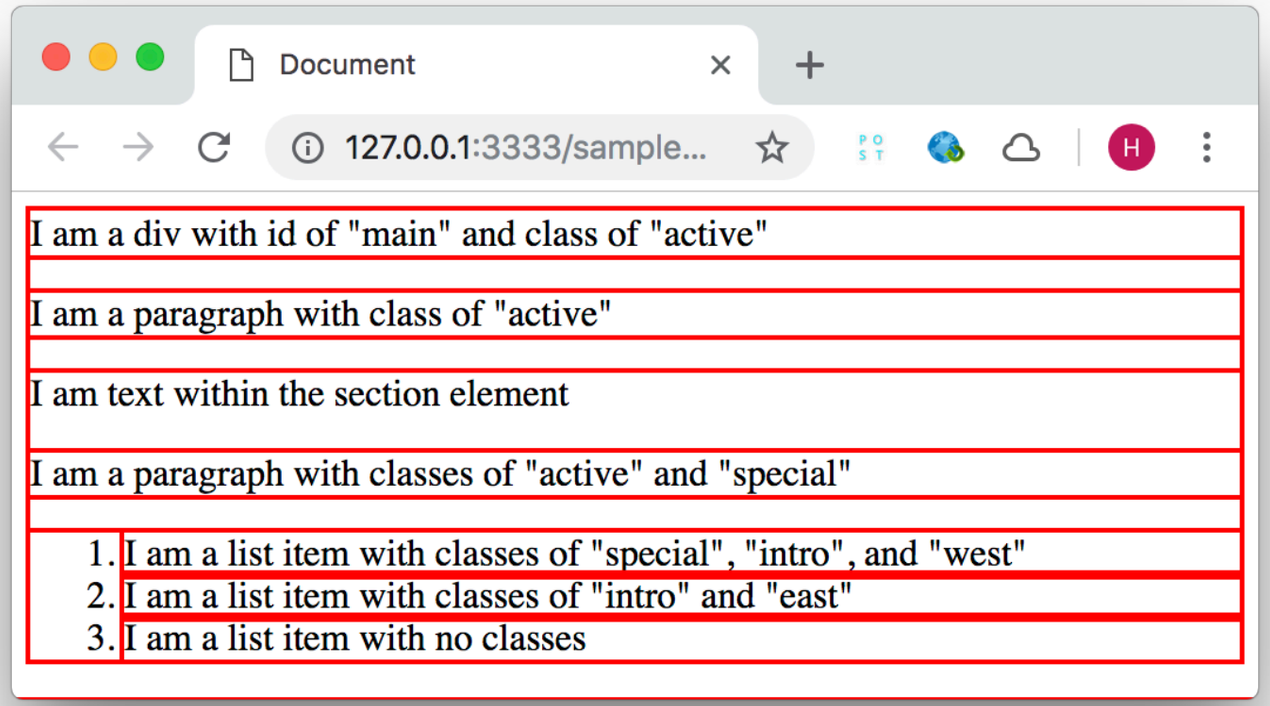
You could have another element with an id, but its value could not be "main".




```
#main {
 outline: 2px solid red;
}
```

Element with an id of "main"

UNIVERSAL SELECTOR




```
*{
 outline: 2px solid red;
}
```

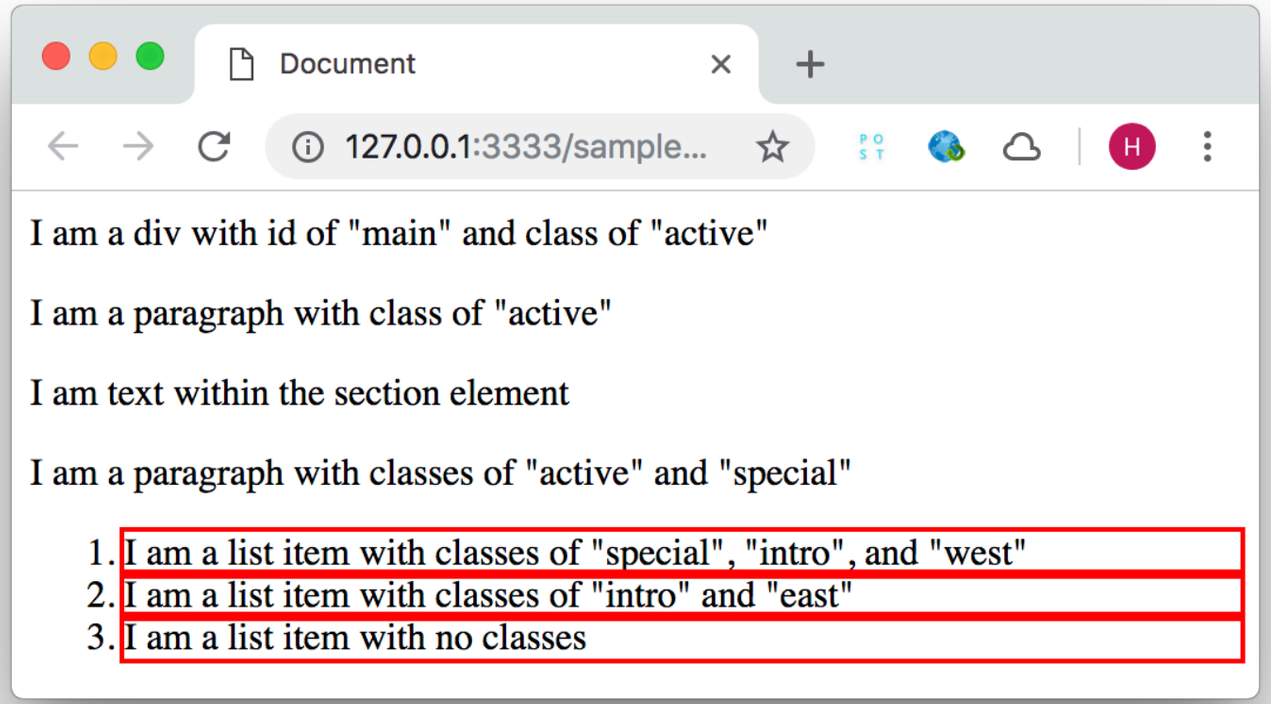
All elements

DESCENDANT SELECTORS

There are multiple ways of getting to the same elements.

As an example, the list items (<li>) can be selected either as a simple type selector or a descendant selector.

As a general strategy, the more specific selector (descendants in this case) is a better plan.




```
section>ol>li{
 outline: 2px solid red;
}
```

 elements that are children of <ol> that, itself, is a child of <section>

PSEUDO-CLASS SELECTORS

One of the more interesting selector strategies is the use of *pseudo-classes*. Unlike an explicit class selector, pseudo-classes reflect the *state* of an element. Some example pseudo-selectors are...

- `:active` — an element activated by the user (such as mouse click on that selector)
- `:first-child` — the first element in a group of sibling elements
- `:focus` — an element (typically a form field) that currently has the focus
- `:hover` — an element the mouse hovers over

The use of pseudo-classes is difficult to represent in Guide format. Check the **Intro to CSS Selectors** video to see the use of pseudo-classes.