51)Using multiple css classes & Combined selectors

Here we saw that w ecan apply multiple classes to parent.like-

<section id="intro" class="main-section highlighted">

<p>This is the intro section.</p>

</section>

In css-

.main-section {

height: 800px;

border: 1px solid #ccc;

padding: 16px;

}

.highlighted {

border: 2px solid #fa923f;

}

Here border defined by second class will win because, that class is defined later in file. if we switch the order(in css file), then border defined by main-section will be applied.

Second thing we learner in this lecture is combined selector-

a.active{

}

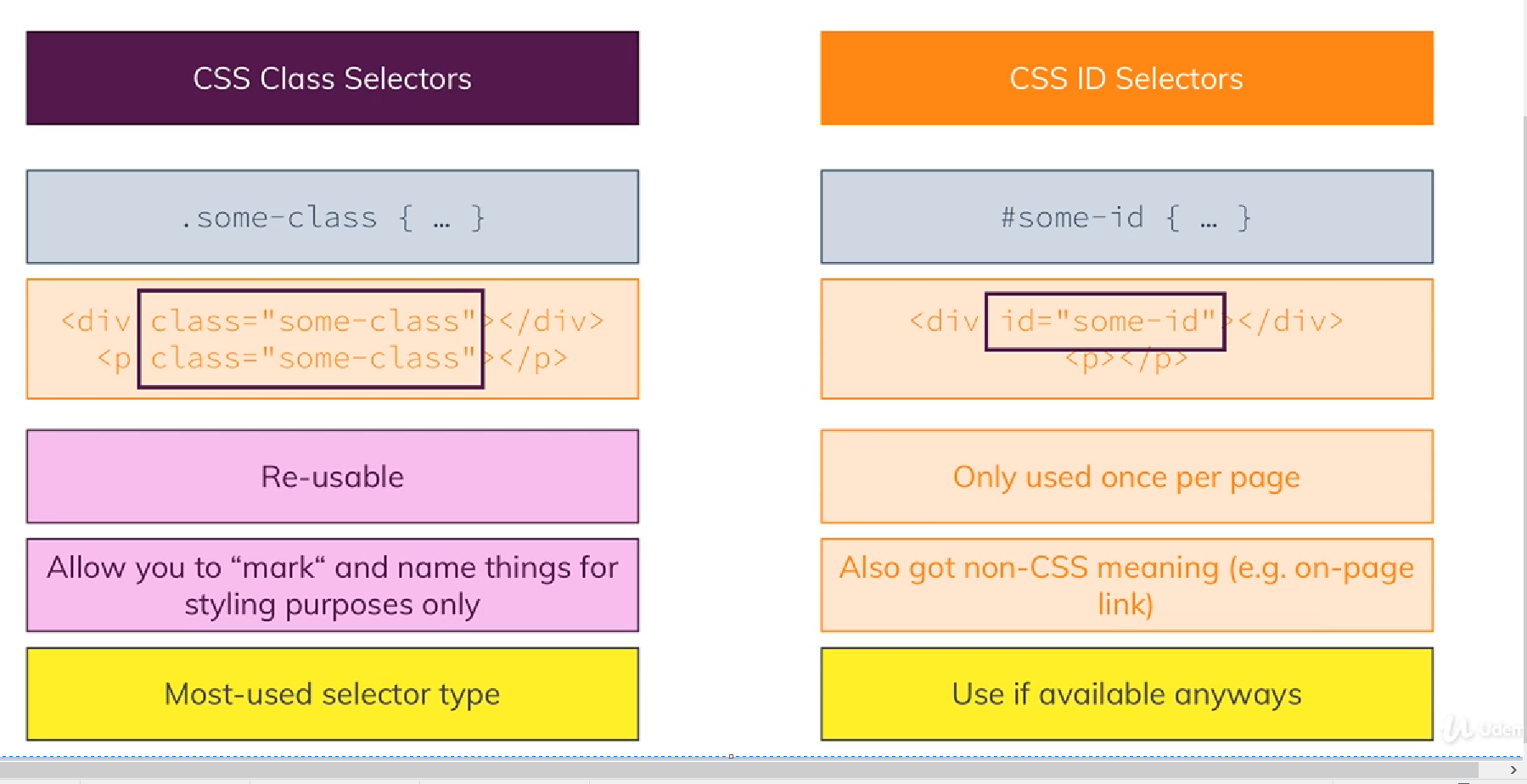
This selector will target the a tags which have active class. If we give space between a and ‘.active’ then it will look for elements with active class and having a tag as parent(direct or indirect). We can do same thing with id like-

a#active{

}

Also this is not limited to tag- class selector combinator. you can have 2 classes, you can have a id and class or tag with 2 classes. You can simply chain selectors into one selector to be more specific about what you want to select.

52)Classes or ID’s



Here we saw that css are used only for styling and it is never wrong to use them. They are also used in conjuction with js.so using classes is never wrong. So they should be your first pick.

ID selector can be decent choice too, though. If you have style, that should be applied only to one element on page, then using id selector, is not a bad idea.however keep in mind that id’s have non css meaning(that link thing). mostly the thing that you can link to id’s.so therefore usin id’s just to apply a style is not really something I can recommended. So use a class to apply style, even if style is to applied be applied only to one element. So use id’s if they semantically make sense, but do not use them just for styling purpose.we can also id’s for styling purpose, as we saw we can use id’s to scroll to a particular part in website.

53)(Not using !Imortant)

See lecture

54. Selecting the Opposite with :not()

See all lectures after that.