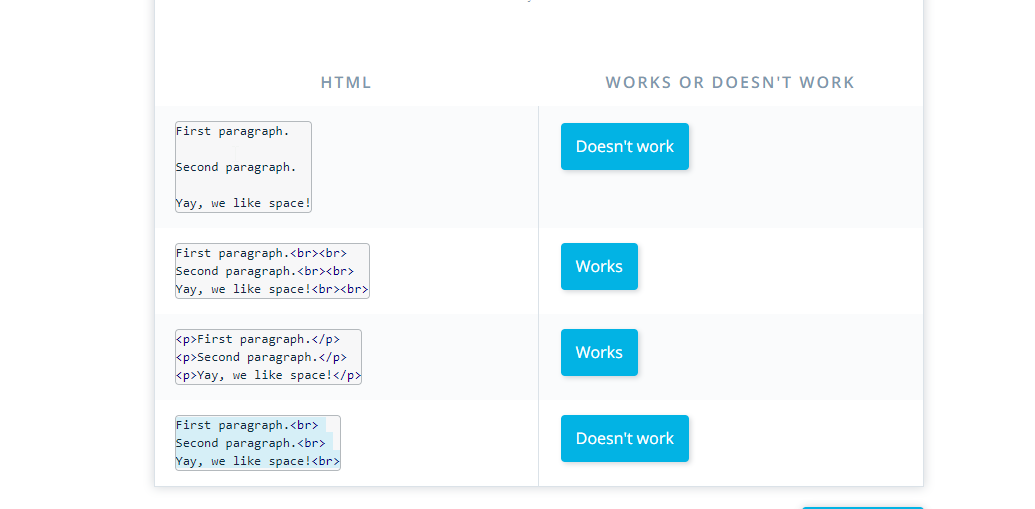
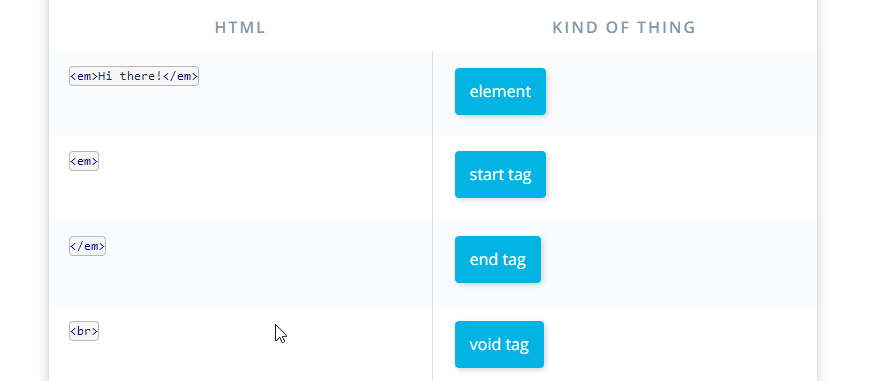
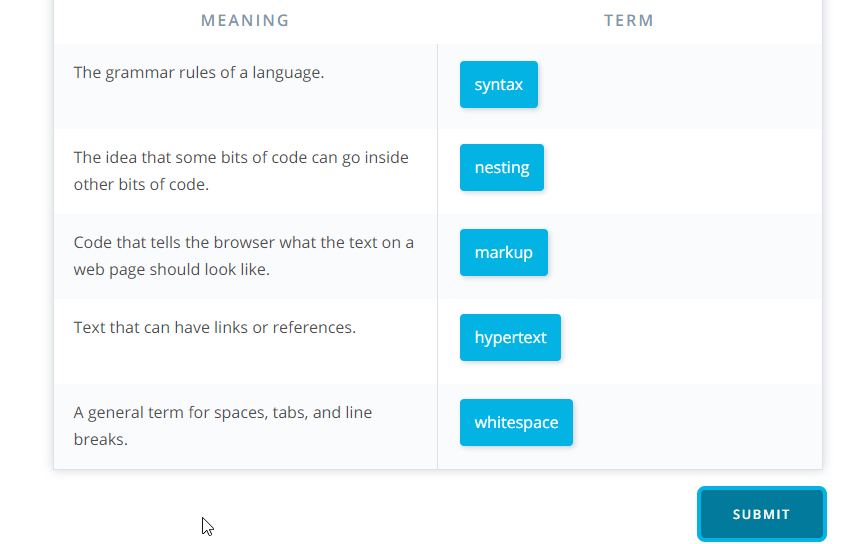
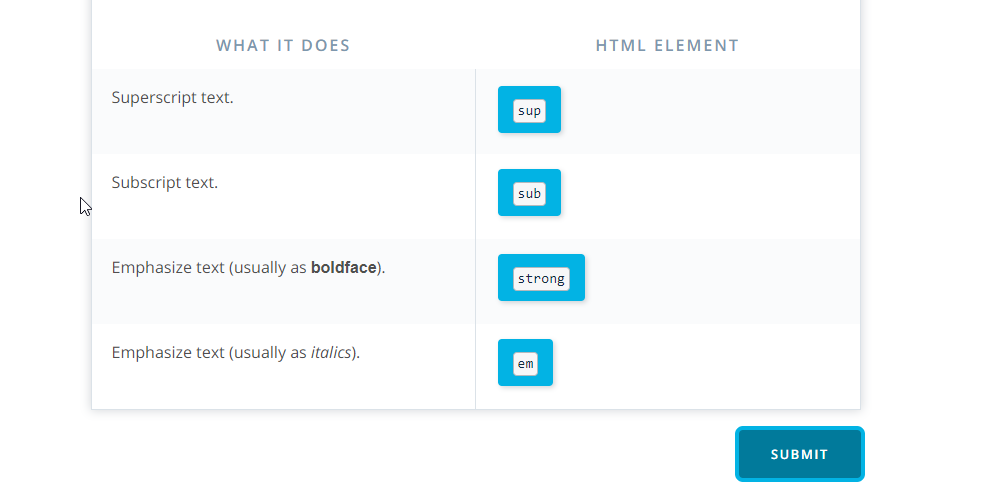
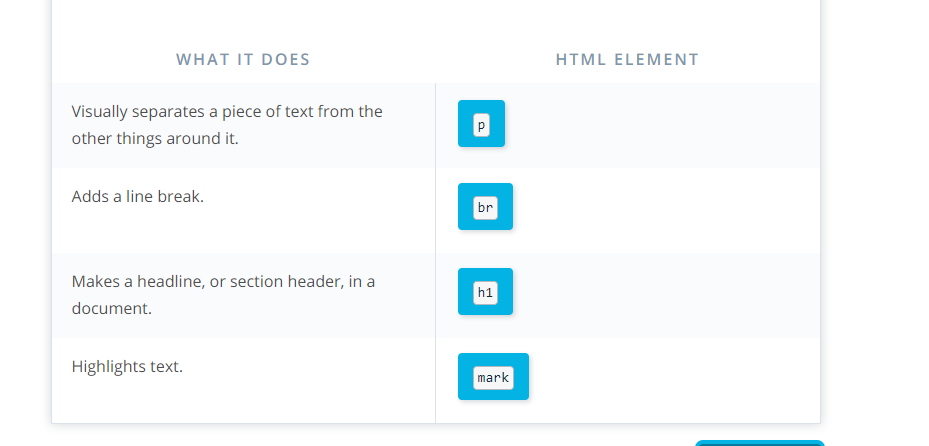
## Fragen und Antworten

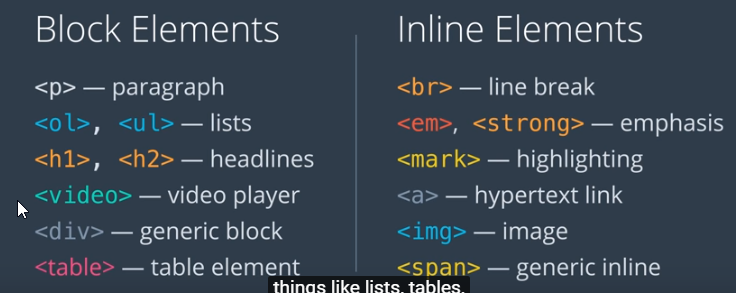


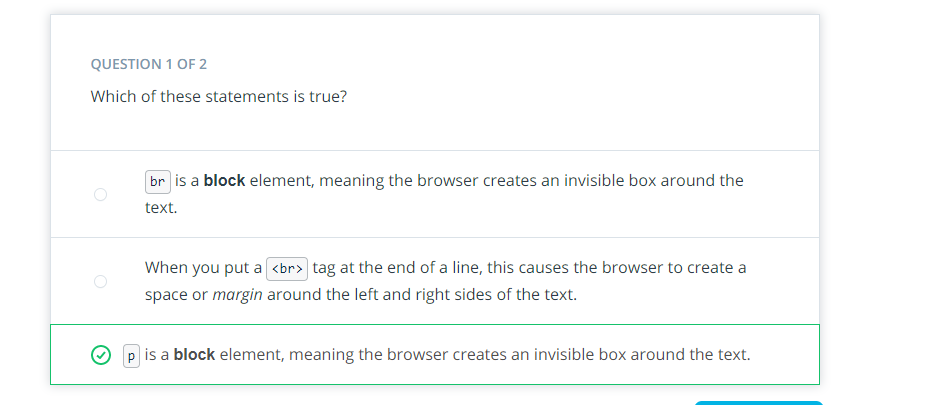


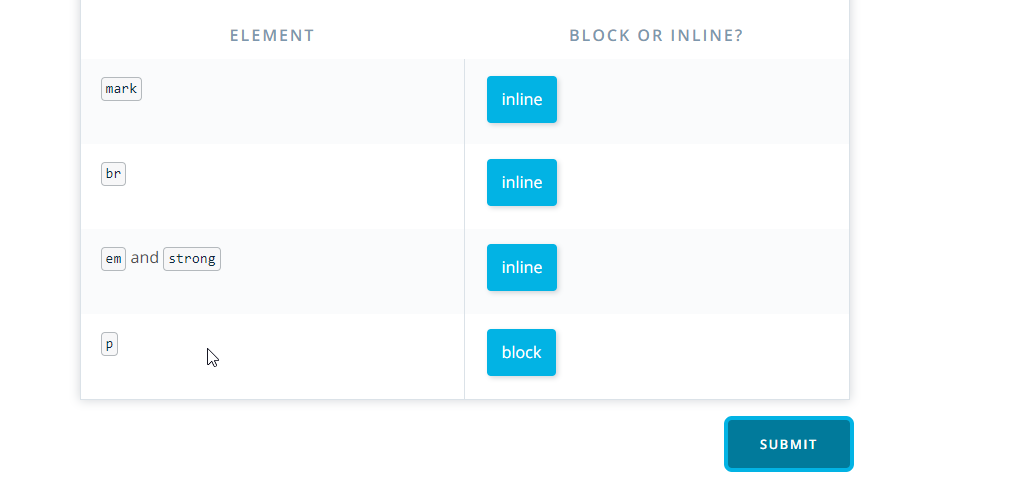


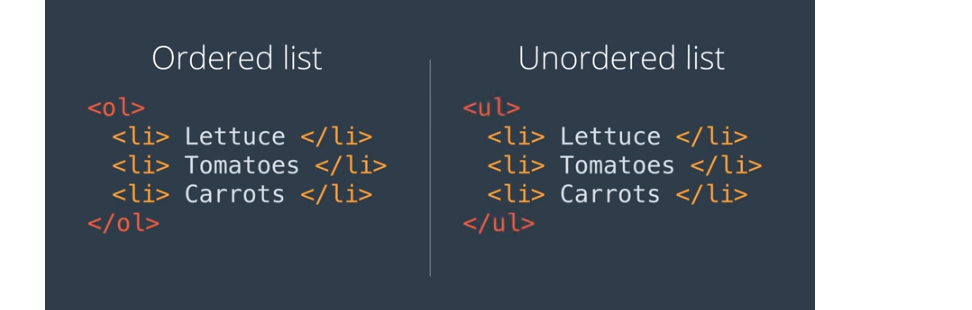


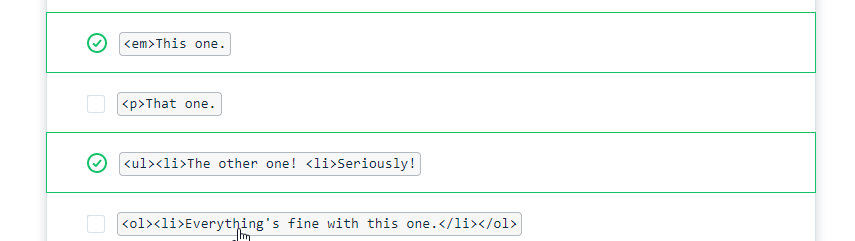


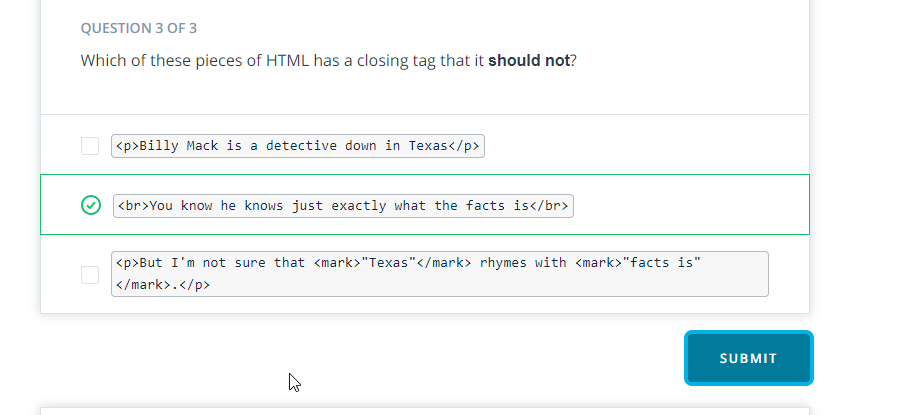
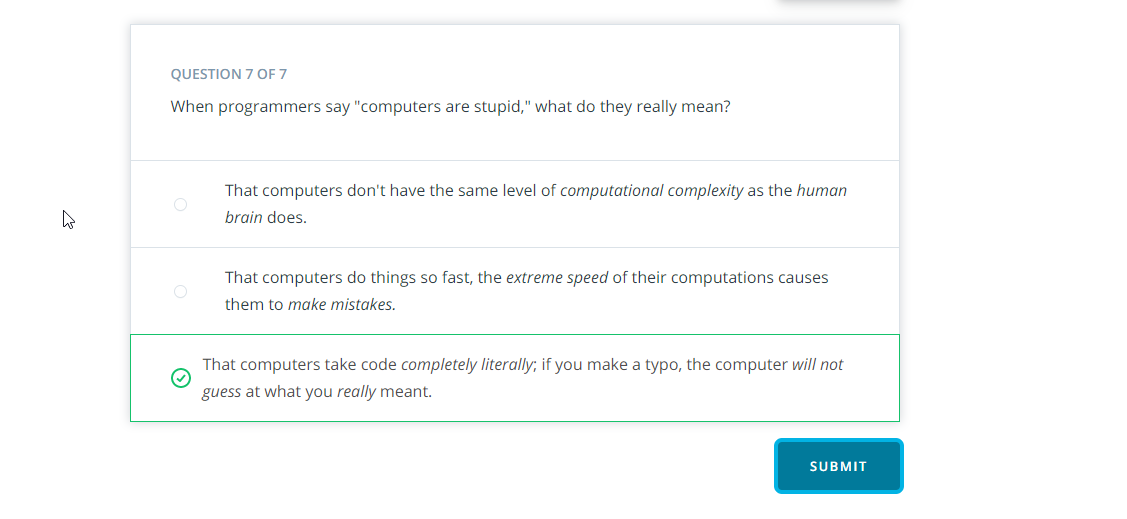


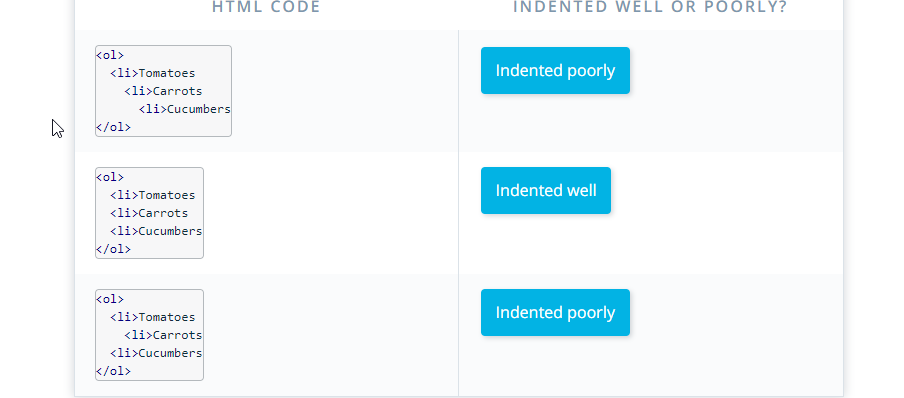


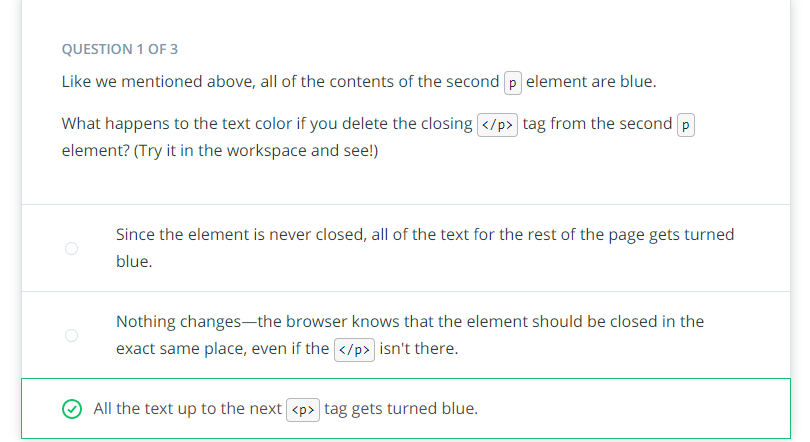


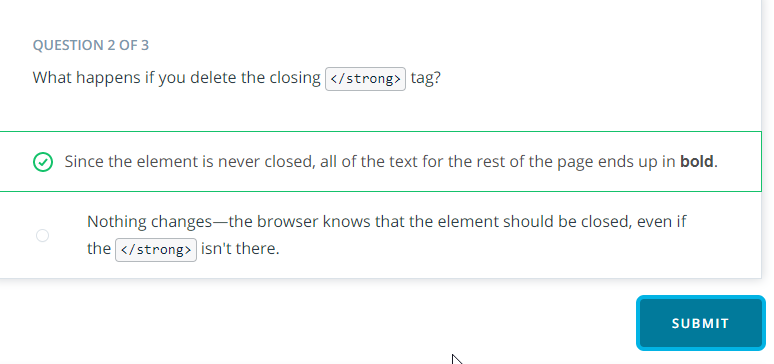


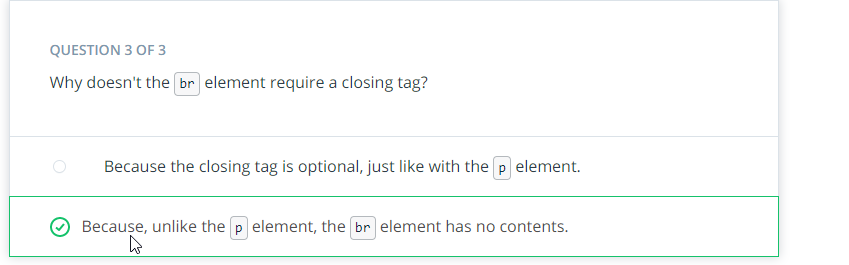


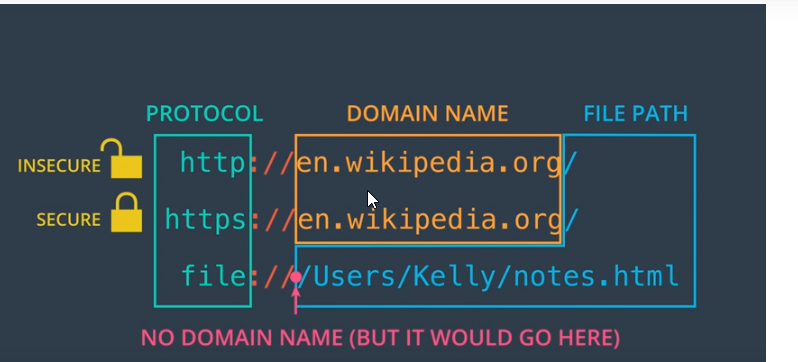


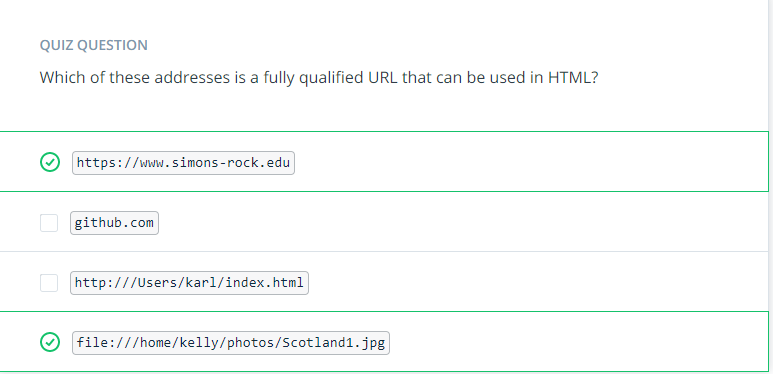


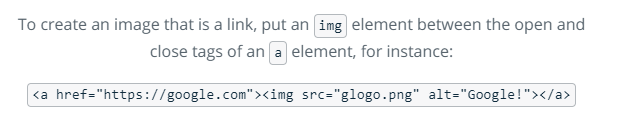


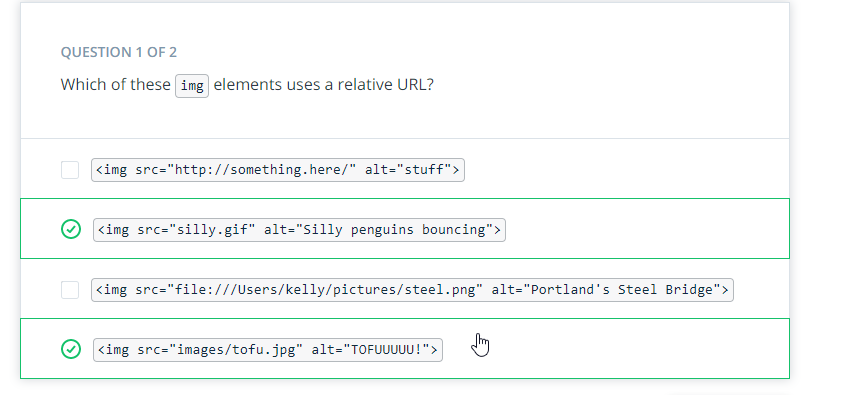


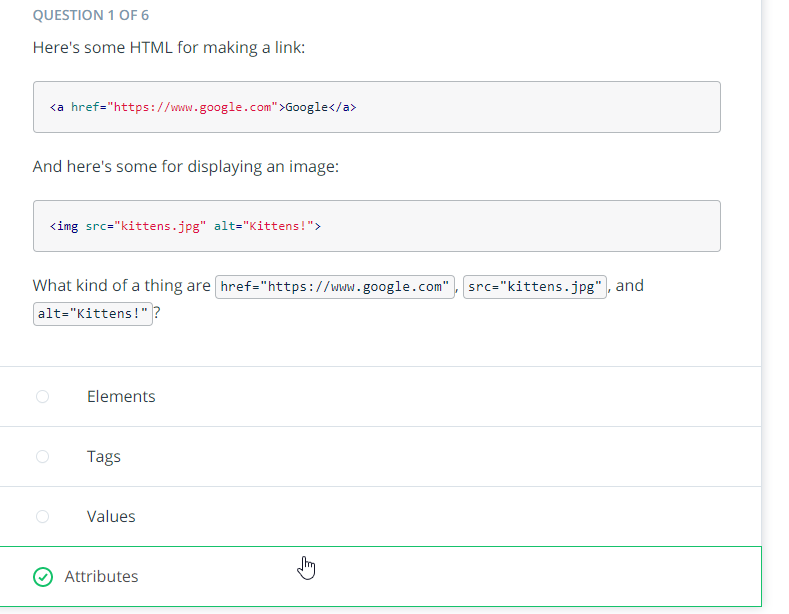


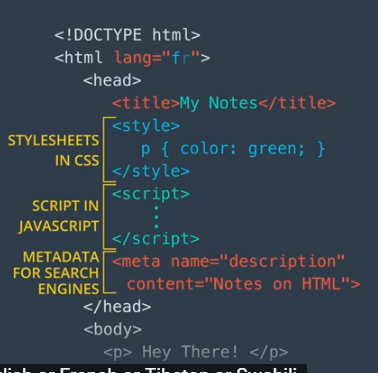


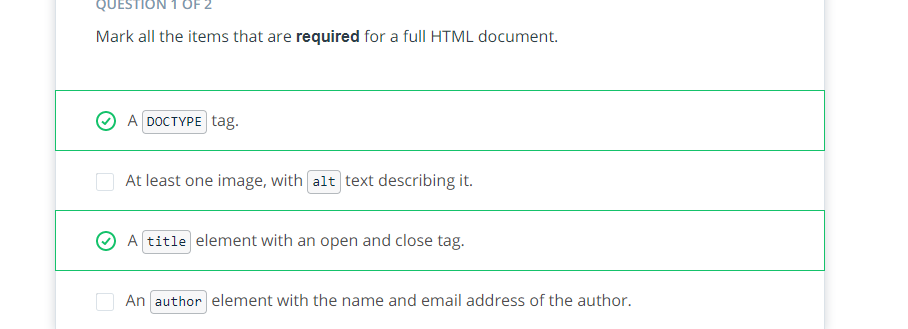


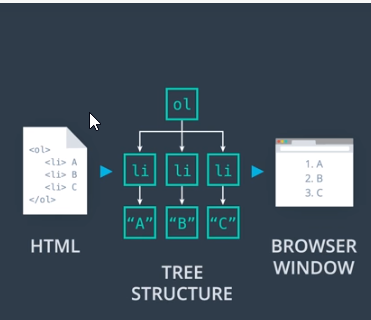


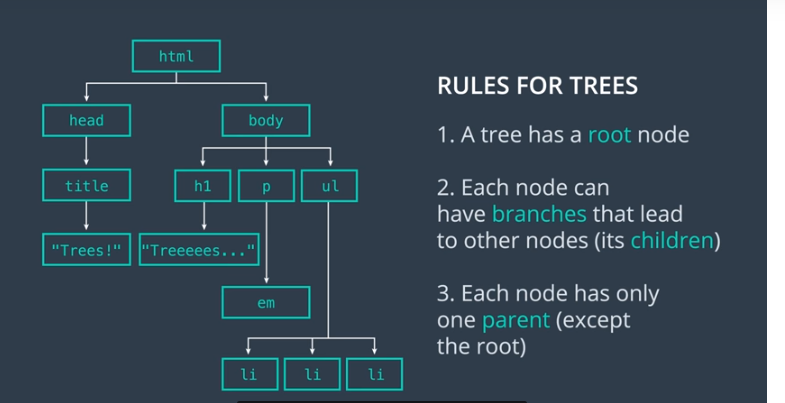


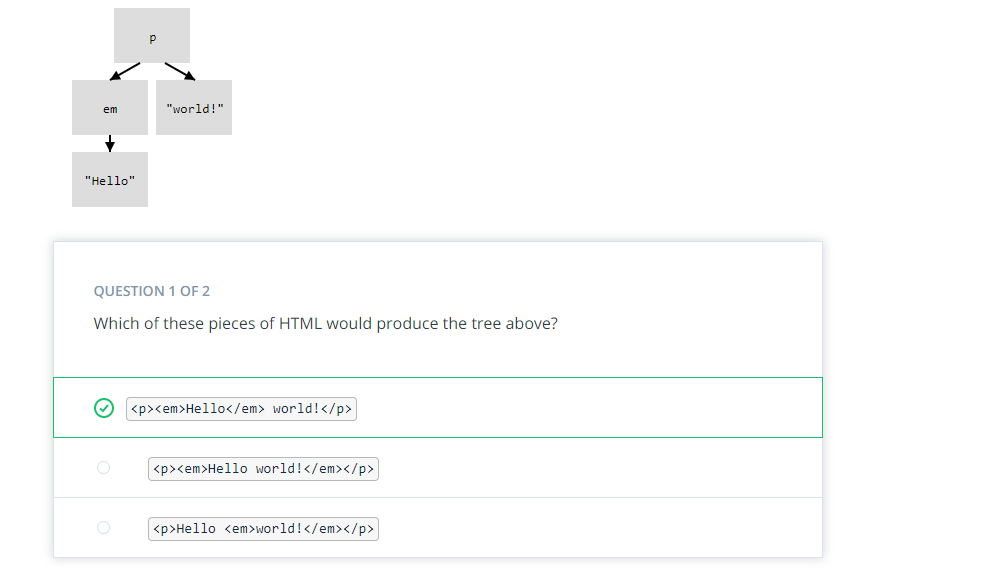


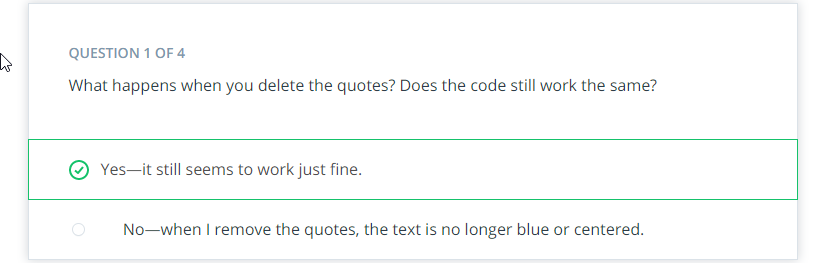


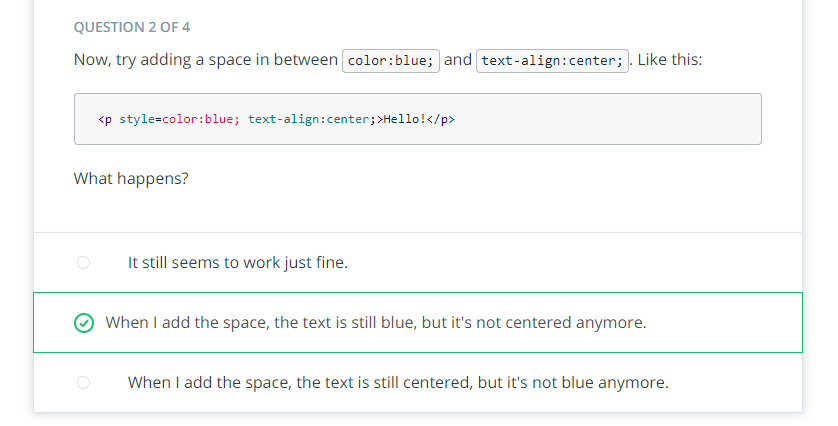


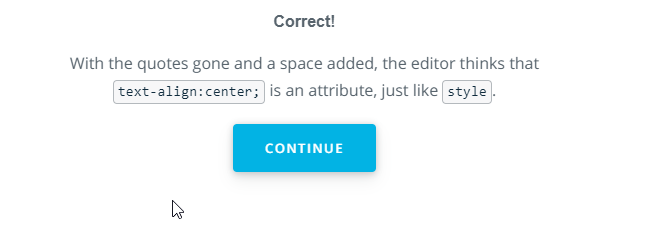


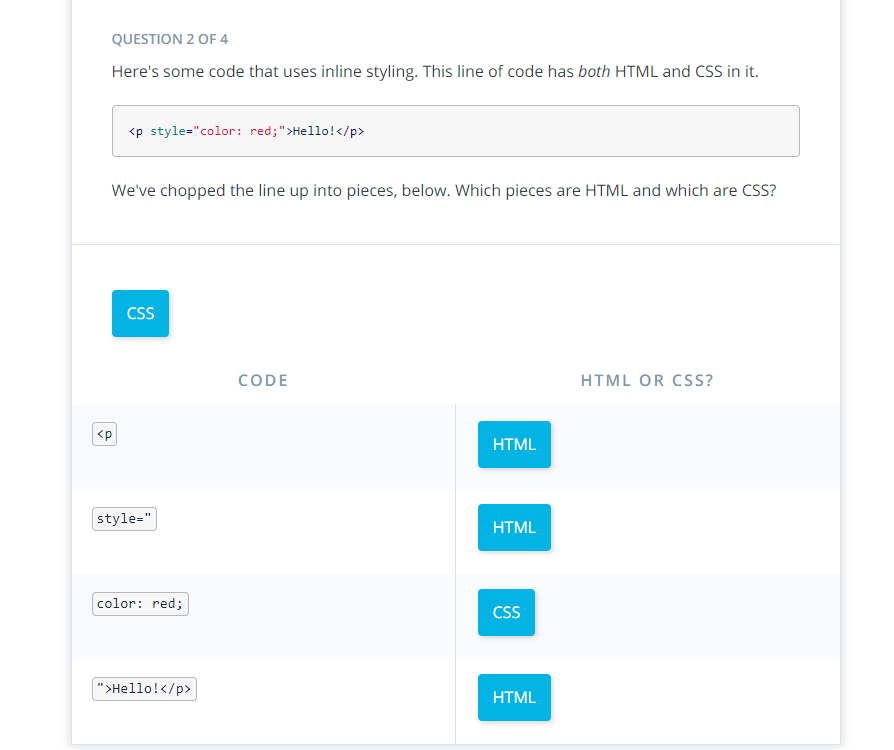


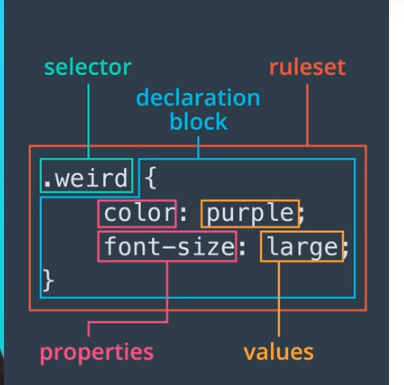






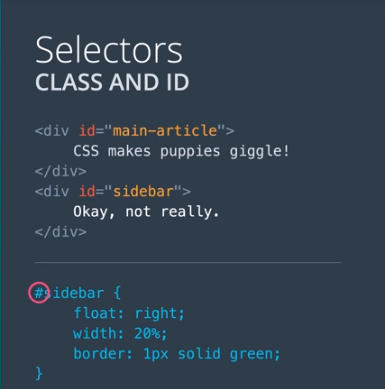








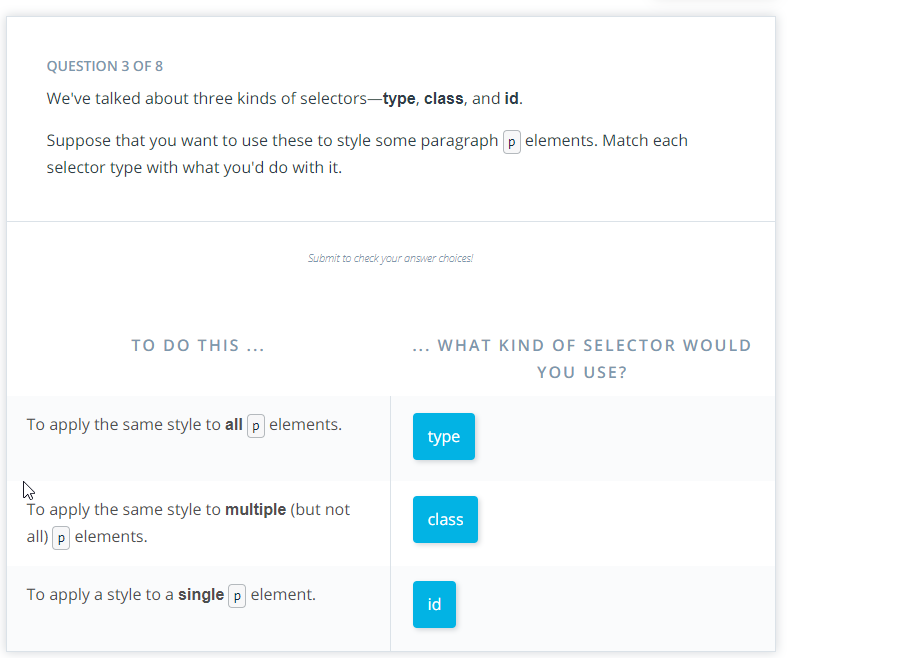


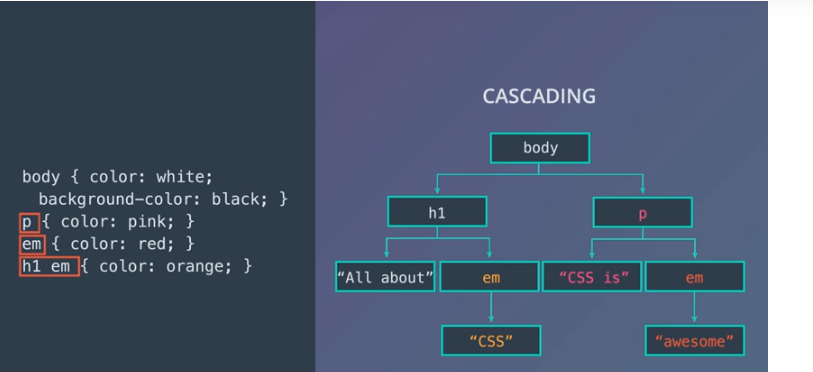






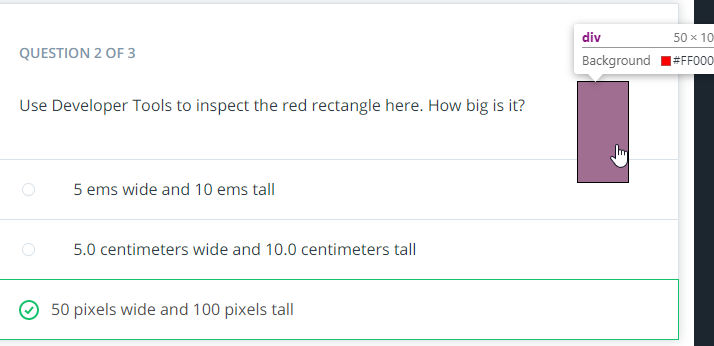
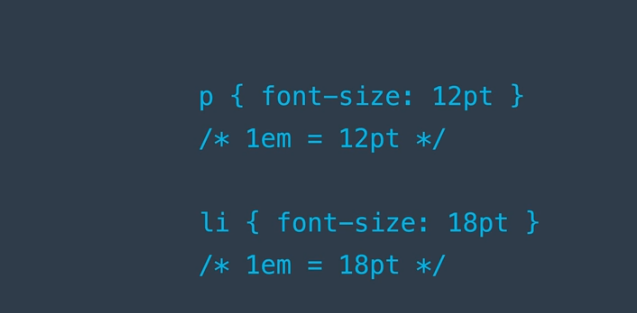




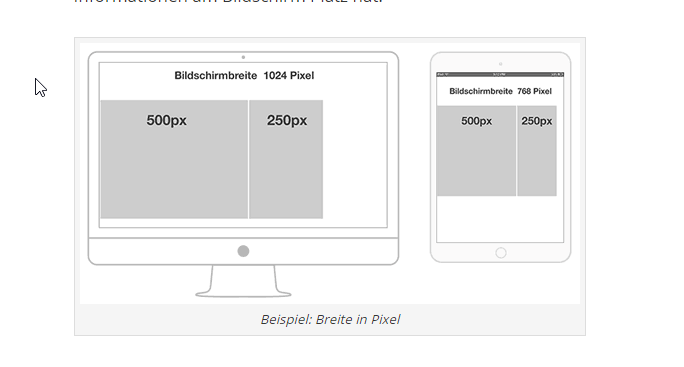




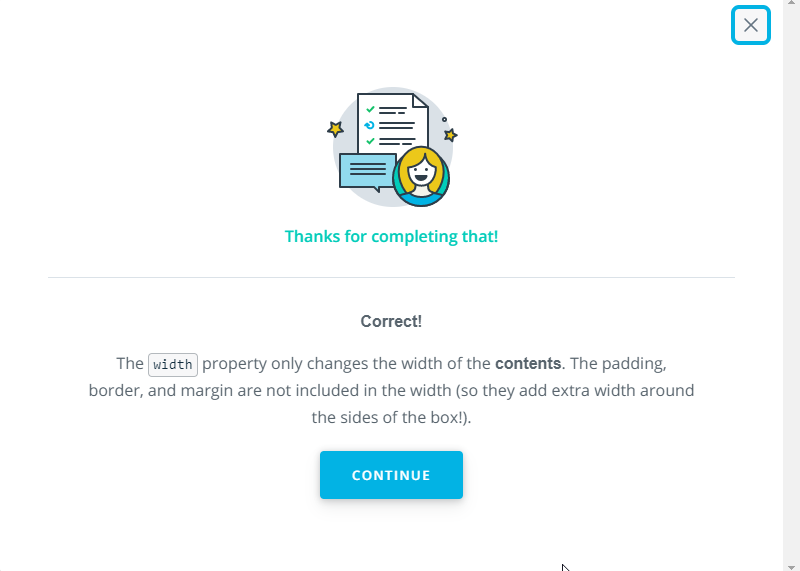


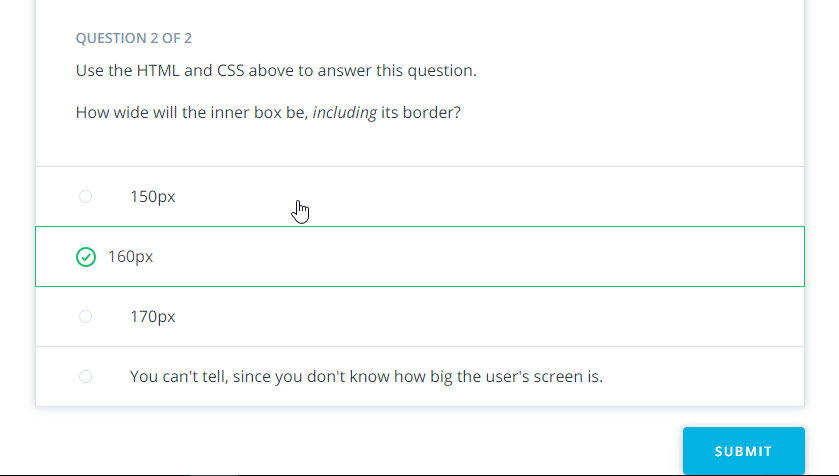


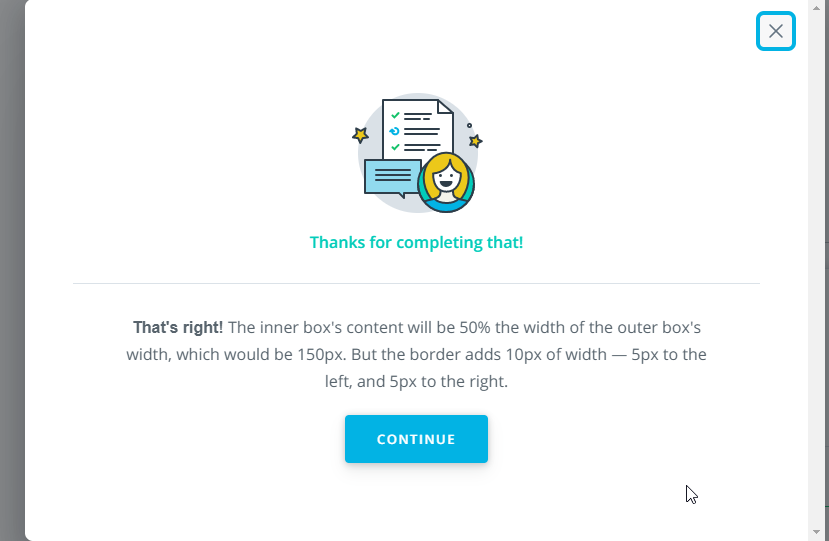


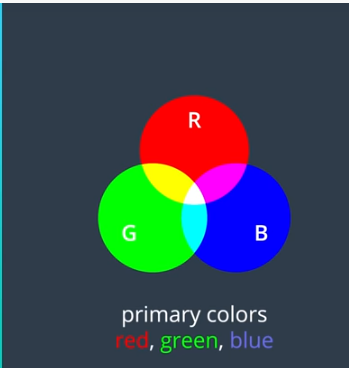


1em = 16 pixel

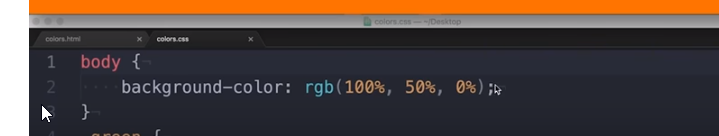


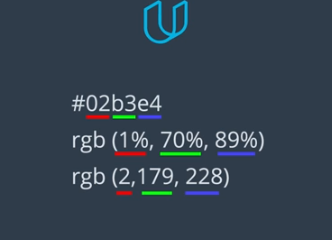


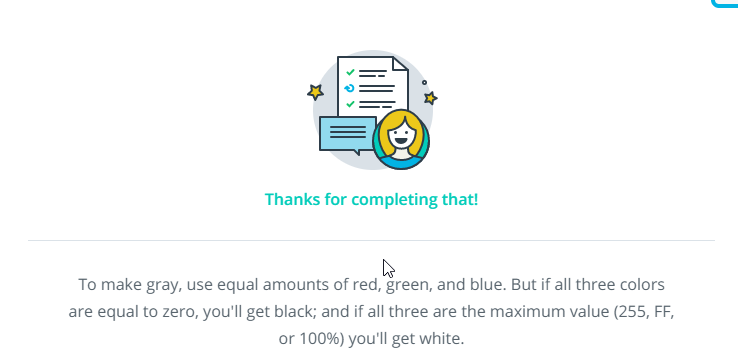


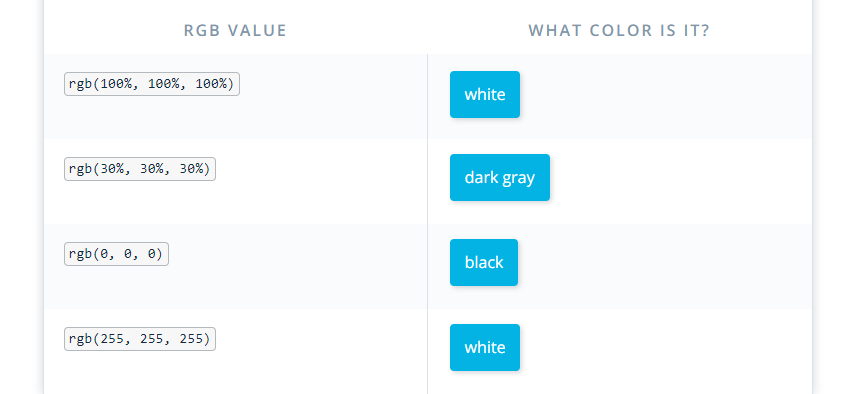


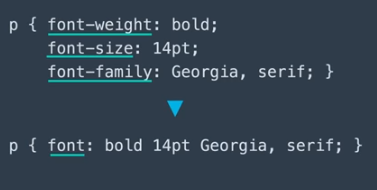


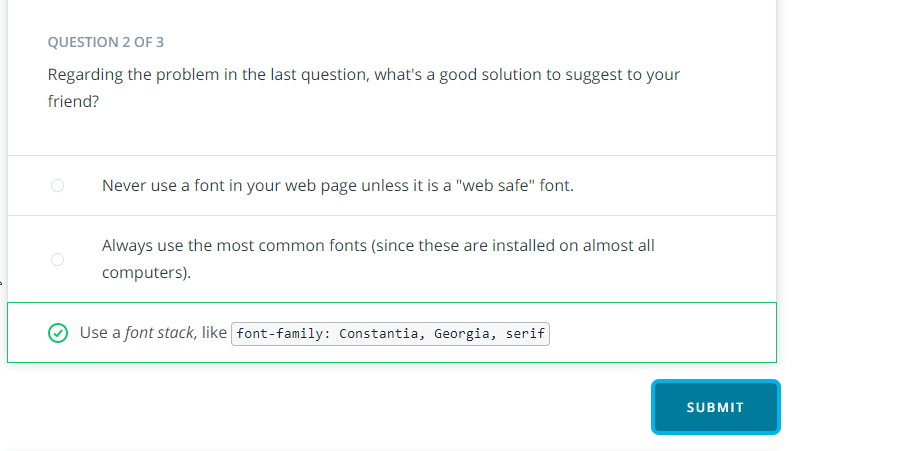


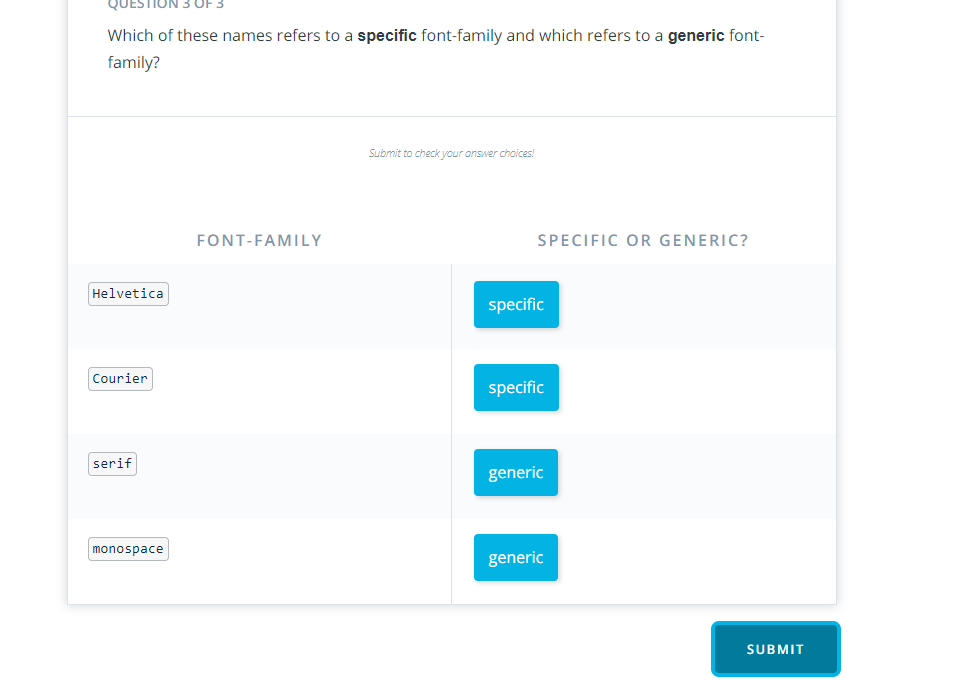


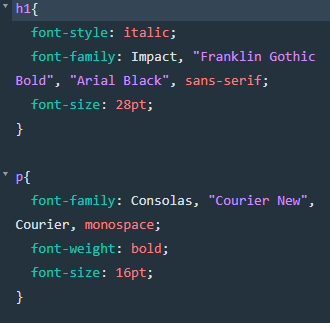


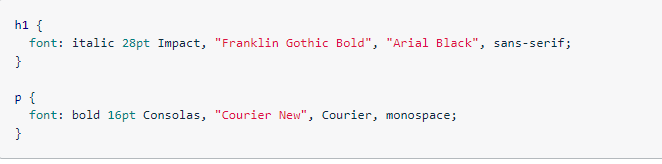








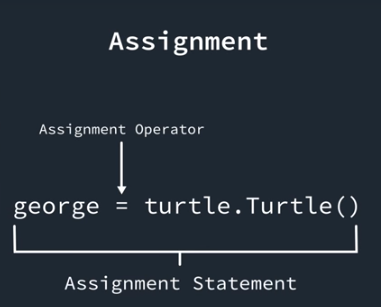




# Flexbox



# Python



## parameters ... or arguments?

In practice, different programmers aren't always consistent when they use the terms "argument" and "parameter"—and it's not uncommon to see the two terms get mixed up. And some programmers avoid the term "argument" altogether, and instead use the terms "formal parameter" and "actual parameter".

It would be nice (and less confusing) if we were all consistent with our terminology! However, the most important thing is not what we call these things, but that we understand **what they are and how they work**: When we call a function, we pass it some inputs (usually called **arguments**)—and these inputs get assigned to variables that are in the definition of that function (usually called **parameters**).

If you understand that underlying core idea, you'll usually be able to tell what someone is talking about from the context—regardless of what terms they use.

# Shell Workshop

