

Properties Worthy of Mention

Here are some behaviors that are worth mentioning.

1. Custom properties are ordinary properties, so they can be declared on any element.

Declare them on a paragraph element, section, aside, root, or even pseudo-elements. They'll work as expected.

```
p {  
  --color: blue  
}  
  
section {  
  --color: #bad  
}  
  
aside {  
  --color: yellow  
}  
  
:root {  
  --color: teal  
}  
  
p:before {  
  --color: red  
}
```




Like normal properties,
They work everywhere :)

2. CSS variables are resolved with the normal inheritance and cascade rules

Consider the block of code below:

```
div {  
  --color: red;  
}  
  
div.test {  
  color: var(--color)  
}  
  
div.ew {  
  color: var(--color)  
}
```

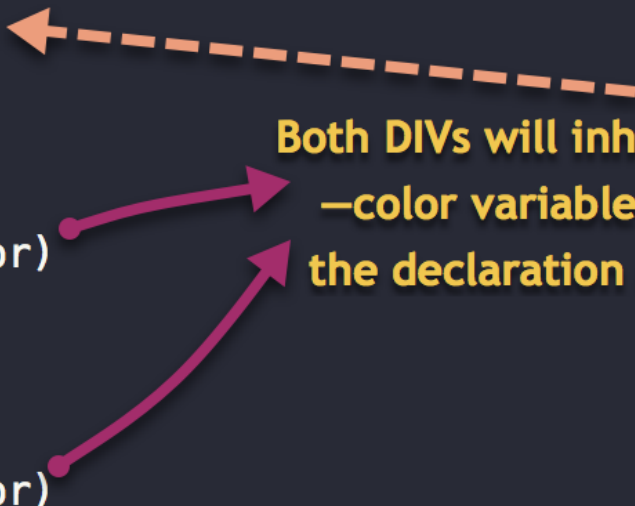
As with normal variables, the `--color` value will be inherited by the divs

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

```
div.test {  
  color: var(--color)  
}
```

```
div.ew {  
  color: var(--color)  
}
```

**Both DIVs will inherit the
--color variable from
the declaration above**



3. CSS variables can be made conditional with `@media` and other conditional rules

As with other properties, you can change the value of a css variable within a `@media` block or other conditional rules.

For example, the following code changes the value of the variable, gutter on larger devices.

```
:root {
  --gutter: 10px
}

@media screen and (min-width: 768px) {
  --gutter: 30px
}
```

```
:root {
  --gutter: 10px
}

@media screen and (min-width: 768px) {
  --gutter: 30px
}
```

I'm 10px here.

Now I'm 30px on larger devices. 🌟

4. CSS variables can be used in HTML's style attribute.

You can choose to set the value of your variables inline, and they'll still work as expected.

```
<!--HTML-->
<html style="--color: red">

<!--CSS-->
body {
  color: var(--color)
```

```
color: var(--color);  
}
```

```
<!--HTML-->
```

```
<html style="--color: red">
```

```
<style>
```

```
  body {
```

```
    color: var(--color)
```

```
  }
```

```
</style>
```

This works!

**You can set variables
inline.**

CSS variables are case sensitive. Be careful with this one. I save myself the stress and write variables in the lower case. Your mileage may vary.

```
/*these are two different variables*/  
:root {  
  --color: blue;  
  --COLOR: red;  
}
```

Be sure to toy with the variable in the playground below. Add some more variables and try out the different behaviors discussed in this lesson.

HTML CSS Output

```
1  div {  
2    --color: teal;  
3    width: 200px;  
4    height: 200px;  
5    display: inline-block;  
6  }  
7  
8  div.test {  
9    background-color: var(--color)  
10 }  
11  
12 div.ew {  
13   background-color: var(--color)
```

output

```
14 }  
15  
16  
17  
18  
19  
20  
21
```

Class is TEST

Class is EW. What you gonna
do about that :)



See you in the next lesson!