

React Component Instantiation

Next, I'll briefly explain JavaScript classes, to help clarify React components. Technically they are not related, but it is a fitting analogy for you to understand the concept of a component.

Classes are most often used in object-oriented programming languages. JavaScript, always flexible in its programming paradigms, allows functional programming and object-oriented programming to co-exist side-by-side. To recap JavaScript classes for object-oriented programming, consider the following *Developer* class:

```
class Developer {  
  constructor(firstName, lastName) {  
    this.firstName = firstName;  
    this.lastName = lastName;  
  }  
  
  getName() {  
    return this.firstName + ' ' + this.lastName;  
  }  
}
```

src/App.js

Each class has a constructor that takes arguments and assigns them to the class instance. A class can also define functions that are associated with a subject (e.g. `getName`), called **methods** or **class methods**.

Defining the Developer class once is just one part; instantiating it is the other. The class definition is the blueprint of its capabilities, and usage occurs when an instance is created with the `new` statement.

```
// class definition  
class Developer { ... }  
  
// class instantiation  
const robin = new Developer('Robin', 'Wieruch');
```

```

console.log(robin.getName());
// "Robin Wieruch"

// another class instantiation
const dennis = new Developer('Dennis', 'Wieruch');

console.log(dennis.getName());
// "Dennis Wieruch"

```

src/App.js

If a JavaScript class definition exists, one can create *multiple* instances of it. It is similar to a React component, which has only *one* component definition, but can have *multiple* instances:

```

  IHDR      (-S  äPLTE""""""""""2PX=r)7;*:>H-BGE8do5Xb6[eK®K`1ML
  IHDR      x@ÍÊ  ePLTE""""""""""2RZNç¹J«3R[J(-)59YÁp0KS4W`Q«ÄL²%
?^q÷ñíÛi.},isæÝ_Ttt0% 1#□/(i□-[□□□è`□è`Ì□ÚiÅðZ□d5□□□?îebZ¿p□i.Úæ□□□iqî□+1°}Â□5
  IHDR      DæAE  APLTE  """"""""""2RZVºÖ_ÔôU·Ñ=r□$()'25]îíC□□
  IHDR  @  @□□  □·□i  □:PLTE  """"""""""
φßqÇ8Û'□mKÊ±mÆJmÜü·yi!è□îªYïuë Äï_Äï?i÷□ý+ð□□ÄA□|□ù{□□'?¿□_En□).□JÊDæ<□
@-φZ\Ts@R*□(□  @□J□□□□u□X/□4J□9□j5·DEµ4kÇ4□&i¥V4Ú□j®□□□_□vsf:àg,□φèBC»i$J□ºíùi□□á□@
-ê>Û□º«φXÖφi}ß"ëÜÑ;□ÄöN'□φvÅý□î,ÿ1  □ëxÄO@&v/Äp_□ö\ô□Çí.□□%+0□□;□□□!□fÊ□|'Ó%Â  JY·O□Â□'

```

Once we've defined a **component**, we can use it like an HTML **element** anywhere in our JSX. The element produces an **instance** of your component, or in other words, the component gets instantiated. It's not much different from a JavaScript class definition and usage.

Exercises:

- Familiarize yourself with the terms *component declaration*, *instance*, and *element*.
- Experiment by creating multiple instances of a List component.
- Think about how it could be possible to give each List component its own `list`.