React Component Instantiation

Learn how to instantiate a React component

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:
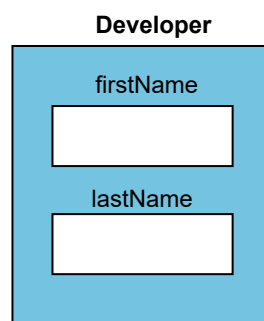
class "Developer" with 2 attributes "firstName" and "lastName"

```
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"
```
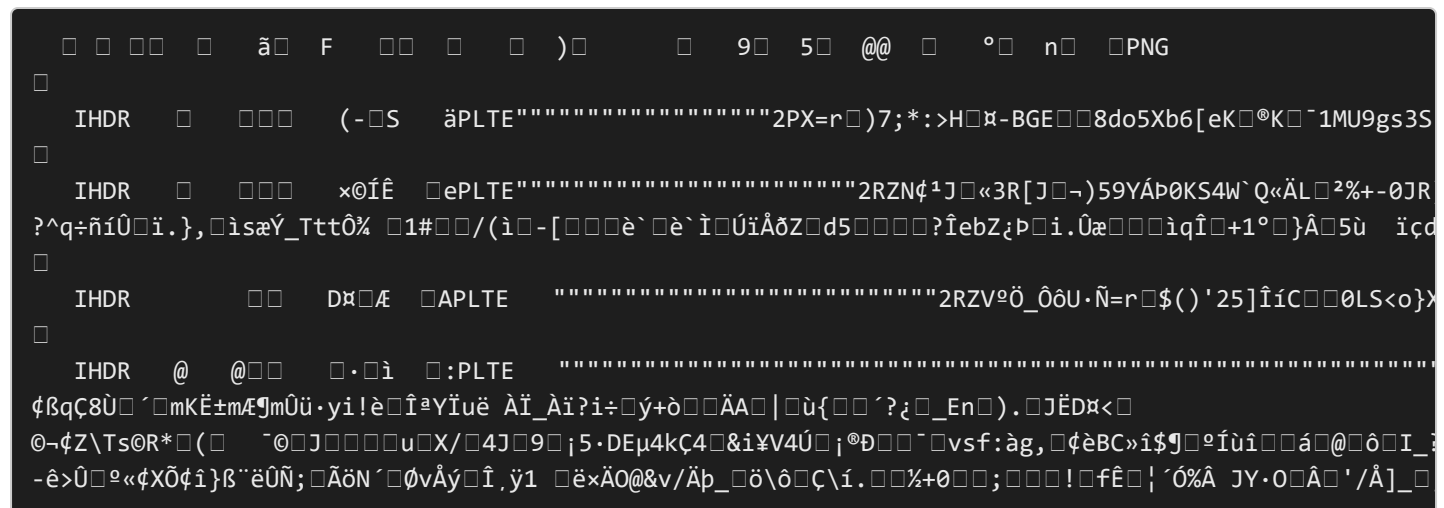
src/App.js

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:

```
    □  □  □□   □    ã□   F    □□   □    □   )□        □    9□   5□   @@   □     °□   n□    □PNG
□
   IHDR    □    □□□    (-□S    äPLTE""""""""""""""""""""2PX=r□)7;*:>H□¤-BGE□□8do5Xb6[eK□®K□¯1MU9gs3S
□
   IHDR    □    □□□    ×©ÍÊ   □ePLTE"""""""""""""""""""""""""2RZN¢¹J□«3R[J□¬)59YÁÞ0KS4W`Q«ÄL□²%+-0JR
?^q÷ñíÛ□ï.},□ìsæÝ_TttÔ% □1#□□/(ì□-[□□□è`□è`Ì□ÚïÅõZ□d5□□□□?ÎebZ¿Þ□i.Ûæ□□□ìqÎ□+1°□}Â□5ù   ïçd
□
   IHDR        □□   D¤□Æ   □APLTE   """"""""""""""""""""""""""""2RZVºÖ_ÔôU·Ñ=r□$()'25]Îí C□□0LS<o}>
□
   IHDR   @   @□□   □·□ì   □:PLTE    """""""""""""""""""""""""""""""""""""""""""""""""""""""""
¢ßqÇ8Ù□´□mKË±mÆ¶mÛü·yi!è□ÎªYïuë ÀÏ_Àï?ì÷□ý+ò□□ÄA□|□ù{□□´?¿□_En□).□JËD¤<□
©¬¢Z\Ts©R*□(□   ¯©□J□□□□u□X/□4J□9□¡5·DEµ4kÇ4□&i¥V4Ú□¡®Ð□□¯□vsf:àg,□¢èBC»î$¶□ºÍùî□□á□@□ô□I_
-ê>Û□º«¢XÕ¢î}ß¨ëÛÑ;□ÃöN´□ØvÅý□Î¸ÿ1 □ë×ÄO@&v/Äþ_□ö\ô□Ç\í.□□%+0□□;□□□!□fÊ□¦´Ó%Â JY·O□Â□'/Å]_□
```

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`.