

Herança - Atributos e Métodos

Herança do construtor

- A herança do construtor permite a utilização do construtor de uma superclasse dentro de uma subclasse (herança do construtor)
- Uma forma simples de fazer isso é utilizando a palavra reservada **super**, dentro do construtor da subclasse

Palavra reservada super

- A palavra reservada `super` pode ser utilizada dentro do construtor da subclasse para chamar o método construtor da superclasse
 - neste caso, deve vir antes da palavra reservada `this`

Palavra reservada super

```
1  class Livro extends Produto {  
2      titulo;  
3      numPag;  
4      constructor(titulo,numPag) {  
5          super('Livro');  
6          this.titulo = titulo;  
7          this.numPag = numPag;  
8          console.log(this.tipo);  
9      }  
10 }
```

Palavra reservada super

- A palavra reservada super também permite acessar os métodos e atributos de uma superclasse dentro de uma subclasse

○

```
1 class Livro extends Produto {
2     titulo;
3     numPag;
4     constructor(titulo, numPag) {
5         super('Livro');
6         this.titulo = titulo;
7         this.numPag = numPag;
8         console.log(`Você instanciou um objeto => ${super.getDados()}`);
9     }
10 }
```

Herança de atributos

- Uma subclasse tem acesso aos atributos públicos de uma superclasse (herança de atributos)
 - posteriormente será explicado um tipo de atributo da superclasse, chamado atributo privado, que a subclasse não tem acesso

Herança de atributos

```
1 class Livro extends Produto {  
2     titulo;  
3     numPag;  
4     constructor(titulo,numPag) {  
5         super('Livro');  
6         this.titulo = titulo;  
7         this.numPag = numPag;  
8         console.log(this.tipo);  
9     }  
10 }
```

o atributo **tipo** foi declarado na superclasse **Produto**, porém por herança ele pode ser acessado pela subclasse **Livro**

Herança de métodos

- Uma subclasse tem acesso aos métodos públicos de uma superclasse (herança de métodos)
 - posteriormente será explicado um tipo de método da superclasse, chamado método privado, que a subclasse não tem acesso

Herança de métodos

```
1 class Livro extends Produto {  
2     titulo;  
3     numPag;  
4     constructor(titulo,numPag) {  
5         super('Livro');  
6         this.titulo = titulo;  
7         this.numPag = numPag;  
8         console.log(this.getDados());  
9     }  
10 }
```

o método `getDados` foi declarado na superclasse **Produto**, porém por herança ele pode ser acessado pela subclasse **Livro**