


() => { }

**6 types of arrow
functions**

1. No Parameters


If the function takes no parameters, you use empty parentheses.



```
1  const greet = () => "Hello!";  
2  console.log(greet());  
3  // Outputs: Hello!
```

2. Single Parameter


If there's only one parameter, parentheses are optional.



```
1  const square = x => x * x;  
2  console.log(square(4));  
3  // Outputs: 16
```

3. Multiple Parameters

If there's only one parameter, parentheses are optional.



```
1  const add = (a, b) => a + b;  
2  console.log(add(2, 3));  
3  // Outputs: 5
```

4. Function Body with Multiple Statements

If the function body has more than one statement, you need to use curly braces and specify the return keyword (if you want to return something).



```
1  const greetPerson = name => {  
2    const greeting = "Hello, " + name + "!";  
3    return greeting;  
4  }  
5  console.log(greetPerson("Alice"));  
6  // Outputs: Hello, Alice!
```

5. Returning Object Literals

When directly returning an object literal, wrap the literal in parentheses to differentiate it from the function block.



```
1  const makePerson = (firstName, lastName) =>
   ({ first: firstName, last: lastName });
2  console.log(makePerson("John", "Doe"));
3  // Outputs: { first: 'John', last: 'Doe' }
```

6. Higher Order Functions and Callbacks

Arrow functions are particularly popular when used as short callbacks.



```
1  const numbers = [1, 2, 3, 4];  
2  const doubled = numbers.map(num => num * 2);  
3  console.log(doubled);  
4  // Outputs: [2, 4, 6, 8]
```