



BABY WOLF CODES

Some random JavaScript

KNICK-KNACKS

```
const fun = a => ({ foo: 'bar' });
```

```
switch (true) {  
  case cond1():  
    // ...  
  case cond2():  
    // ...  
}
```

01



The default parameter value can use the parameters declared before it



```
function test(x, y, z = x + y) {  
  console.log(z);  
}
```

```
test(1, 1, 1); // → 1  
test(1, 2);    // → 3
```

02

...

Scope variables inside case using blocks



```
switch (condition) {  
  case 1: {  
    const message = 'Hello 1';  
    // ...  
  }  
  case 2: {  
    const message = 'Hello 2';  
    // ...  
  }  
}
```

without this, an error is thrown

03



Arrow functions can return an Object, without using a body by wrapping the Object in parantheses



```
const myFun = x => ({ double: x * 2 });
```

```
myFun(1); // → { double: 2 }
```

04



Setting switch expression to true lets us branch by evaluating different conditions




```
switch (true) {  
  case score > 90: return 'A';  
  case score > 80: return 'B';  
  case score > 70: return 'C';  
  default: return 'D';  
}
```


05

...

You can make an event be handled only once by setting once to true



```
element.addEventListener('click', () => {  
  console.log('Element was clicked');  
}, { once: true });
```

06

...

We can destructure an Object property into multiple variables



```
const obj = { val: 1 };

const { val: a, val: b, val } = obj;

console.log(a); // 1
console.log(b); // 1
console.log(val); // 1
```



BABY WOLF CODES

Thanks for reading!

💬 SHARE YOUR
THOUGHTS



If you find my content helpful,
consider donating by clicking on
this highlight in my profile



Support