



BABY WOLF CODES

Some JavaScript Fun Facts



01

The new `at()` method for arrays allows you to access elements from either end easily



```
const arr = ['a', 'b', 'c', 'd'];  
  
arr.at(0); // 'a'  
arr.at(1); // 'b'  
arr.at(-1); // 'd'  
arr.at(-3); // 'b'
```

02

The Date object is smart enough to automatically adjust any overflows in month, date or time values



```
const foo =  
  new Date('December 31, 2022 00:00:00');  
  
foo.setDate(foo.getDate() + 1);
```

foo

```
// -> January 1, 2022
```

03


Underscore can be used as separator for numbers. This is useful for large values to improve readability.



```
const largeNum = 1_000_000;
```

04

Since Strings are iterables, they can be split into an array of characters by using the spread operator



```
const arr = [...'hello'];  
arr  
// -> ['h', 'e', 'l', 'l', 'o']
```

05


The Performance API can be used to calculate time between two events precisely (useful for benchmarking)



```
const start = performance.now();  
  
// ...  
  
const duration = performance.now() - start;  
// Precise time duration
```

06

The classic for loop can be prefixed with `await` to iterate over an array of promises, but wait for each promise to resolve before moving to the next



```
async function test() {  
  const promises = [ /* */ ];  
  for await (const result of promises) {  
    // ...  
  }  
}
```

07

The following syntax provides both the index and value when iterating over an array



```
const arr = ['foo', 'bar', 'baz'];  
  
for (const [key, value] of arr.entries()){  
  console.log(key, value);  
}
```


08

Any piece of code inside a Class is automatically evaluated in strict mode, even if there is no use `strict`



```
class MyClass {  
    // Evaluation here is strict  
}
```



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