1. Arrow function:

- Keep the “this” of parent scope

1. String template:

- String.raw

var text = String.raw`The "\n" newline won't result in a new line. It'll be escaped.`

1. Destructing

**var** robotA = { name: "Bender" };

**var** robotB = { name: "Flexo" };

**var** { name: nameA } = robotA;

**var** { name: nameB } = robotB;

console.log(nameA); // "Bender"

console.log(nameB); // "Flexo"

**var** { foo, bar } = { foo: "lorem", bar: "ipsum" };

console.log(foo); // "lorem"

console.log(bar); // "ipsum"

function greet ({ age, name:greeting='she' })

{

console.log(`${greeting} is ${age} years old.`)

}

greet({ name: 'nico', age: 27 }) // <- 'nico is 27 years old' greet({ age: 24 }) // <- 'she is 24 years old'

function getUrlParts (url)

{

var magic = /^(https?):\/\/(ponyfoo\.com)(\/articles\/([a-z0-9-]+))$/ return magic.exec(url)

}

var parts = getUrlParts('http://ponyfoo.com/articles/es6-destructuring-in-depth')

var [,protocol,host,pathname,slug] = parts

console.log(protocol) // <- 'http'

console.log(host) // <- 'ponyfoo.com'

console.log(pathname) // <- '/articles/es6-destructu..-in-depth'

console.log(slug) // <- 'es6-destructuring-in-depth'

1. Rest & default parameters: …

**function** **animalSentenceFancy**(animals2="tigers", animals3=(animals2 == "bears") ? "**sealions**" : "**bears**")

{

**return** `Lions and ${animals2} and ${animals3}! Oh my!`;

}

function sum (multiplier, base, ...numbers)

{

var sum = numbers.reduce((accumulator, num) => accumulator + num, base)

return multiplier \* sum

}

var total = sum(2, 6, 10, 8, 9)

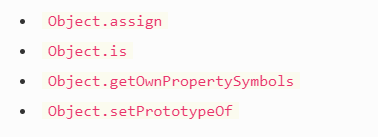
console.log(total) // <- 66

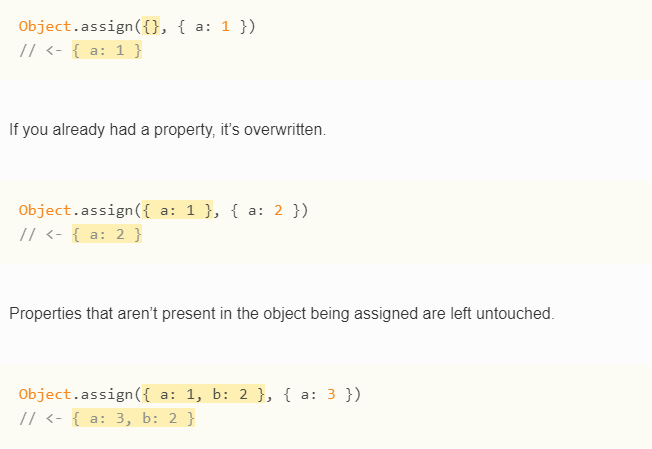
1. Object literal

var foo = 'bar'

var baz = { [foo]: 'ponyfoo' }

console.log(baz) // <- { bar: 'ponyfoo' }





1. Generator

function\* idMaker() {

var index = 0;

while (index < 3)

yield index++;}

var gen = idMaker();

console.log(gen.next().value); // 0

console.log(gen.next().value); // 1

console.log(gen.next().value); // 2

console.log(gen.next().value); // undefined

function\* yieldAndReturn() {

yield "Y";

return "R";

yield "unreachable";}

var gen = yieldAndReturn()

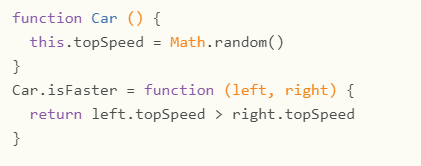
console.log(gen.next()); // { value: "Y", done: false }

console.log(gen.next()); // { value: "R", done: true }

console.log(gen.next()); // { value: undefined, done: true }

1. Class

ES5:



ES6:

