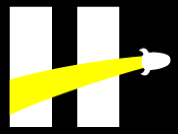
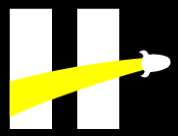
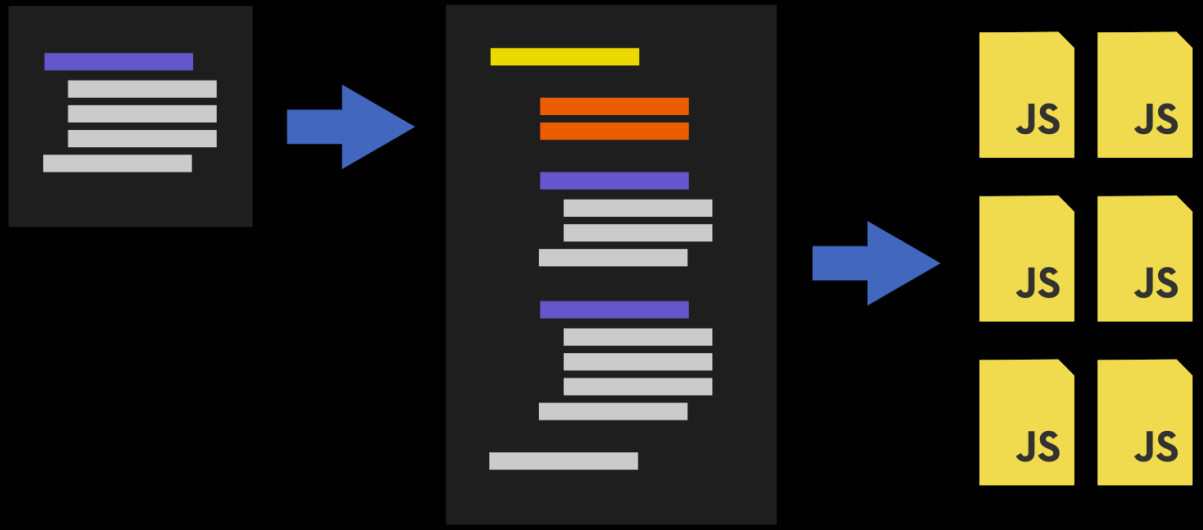
HENRY



MODULOS

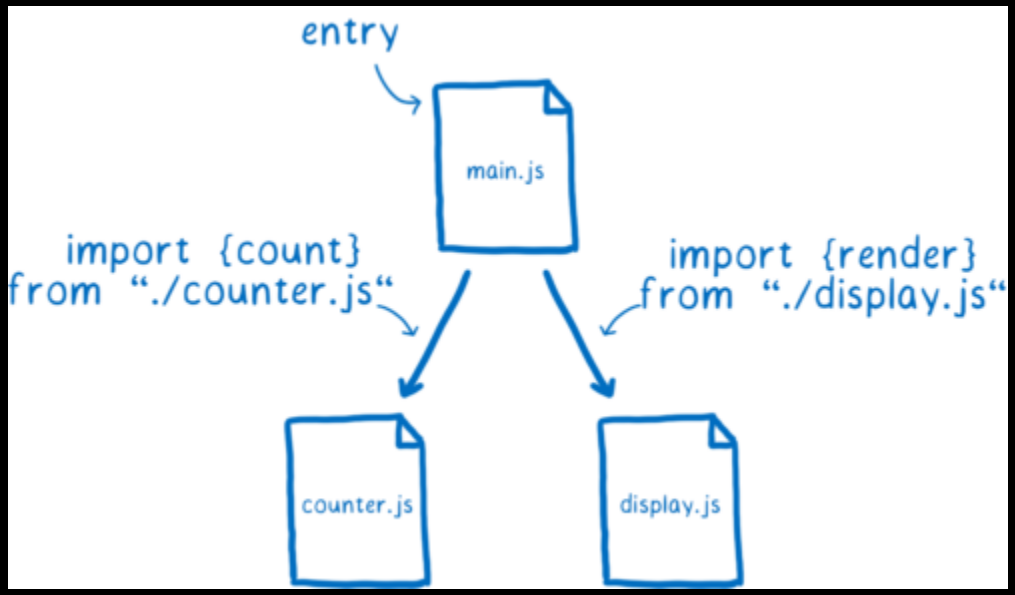


MODULOS

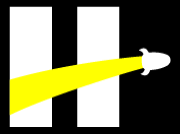


Un **Módulo** es un pedazo de código que cumple una tarea específica y que indica sobre qué piezas de código depende (*dependencias*).

MODULOS



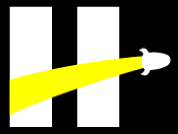
Dependencias



MODULOS

Patrón de módulos IIFE

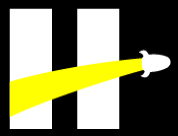
```
1  const weekDay = function() {
2    const names = ["Domingo", "Lunes", "Martes", "Miercoles",
3                   "Jueves", "Viernes", "Sabado"];
4    return {
5      name: function name(number) { return names[number]; },
6      number: function number(name) { return names.indexOf(name); }
7    };
8  }();
9
10 console.log(weekDay.name(weekDay.number("Domingo")));
```



CommonJS

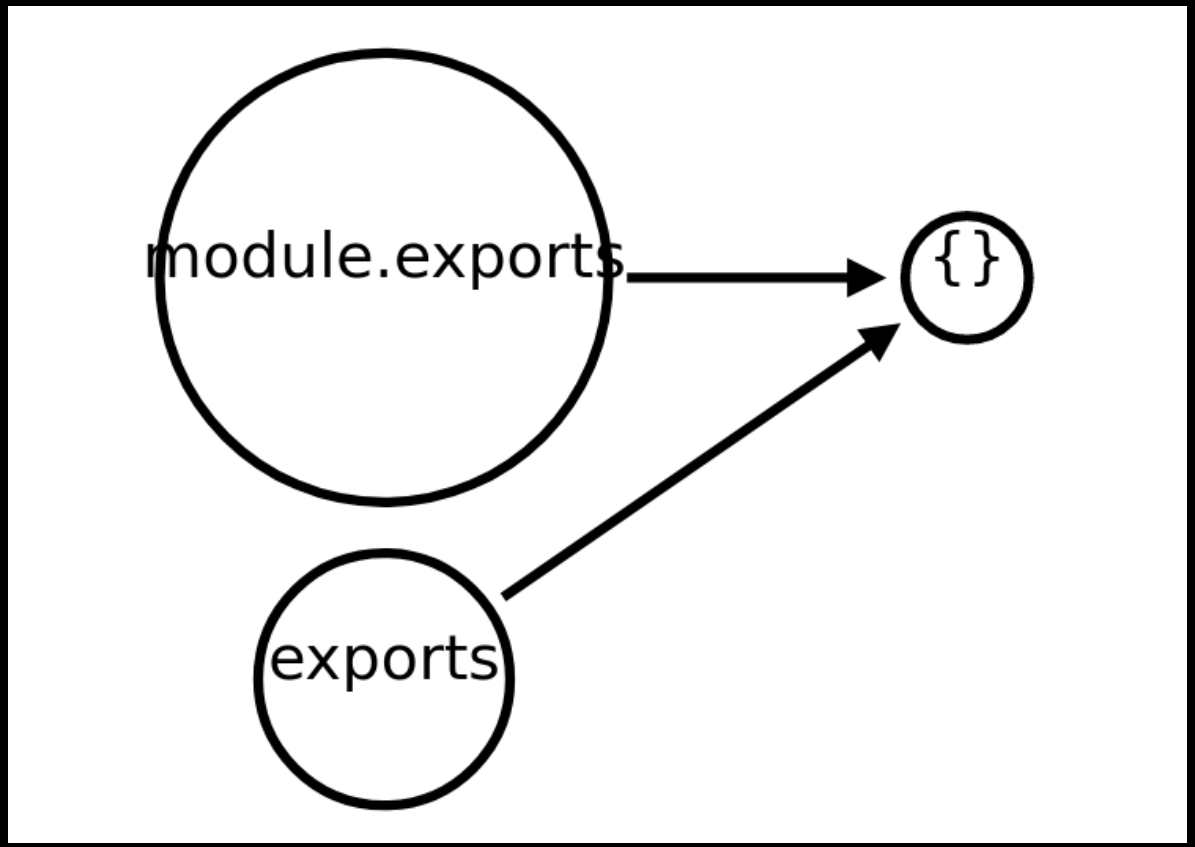
```
1 // WeekDays.js
2
3 var names = ["Domingo", "Lunes", "Martes", "Miercoles",
4                 "Jueves", "Viernes", "Sabado"];
5
6 exports.name = function name (number) { return names[number]; };
7 exports.number = function number(name) { return names.indexOf(name); };
```

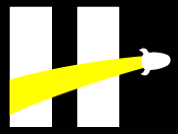
```
1 // index.js
2
3 var weekDays = require('./WeekDays.js');
4
5 console.log(weekDays.name(weekDays.number("Domingo")));
```



module.exports vs exports

Inicialmente



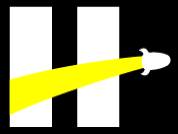


module.exports vs exports

module.exports

```
// module1.js

module.exports.a = function() {
    console.log("a");
}

module.exports.b = function() {
    console.log("b");
}
```

```
// module2.js

var example = require('./module1.js');

example.a(); // "a"

example.b(); // "b"
```

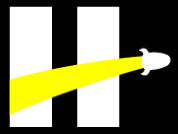


module.exports vs exports

module.exports

```
// module1.js

module.exports.a = function() {
    console.log("a");
}

module.exports.b = function() {
    console.log("b");
}

module.exports = function extra() {
        console.log("Extra function");
}
```

```
// module2.js

var example = require('./module1.js');

example.a(); // TypeError: example.a is not a function

example.b(); // TypeError: example.a is not a function

example.extra(); // TypeError: prueba.extra is not a function

example(); // "Extra function"
```

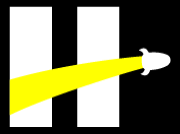


module.exports vs exports

exports

```
// module1.js

exports.a = function() {
    console.log("a");
}
exports.b = function() {
    console.log("b");
}
```

```
// module2.js

var example = require('./module1.js');

example.a(); // "a"

example.b(); // "b"
```

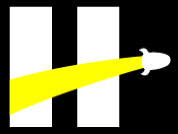


module.exports vs exports

exports

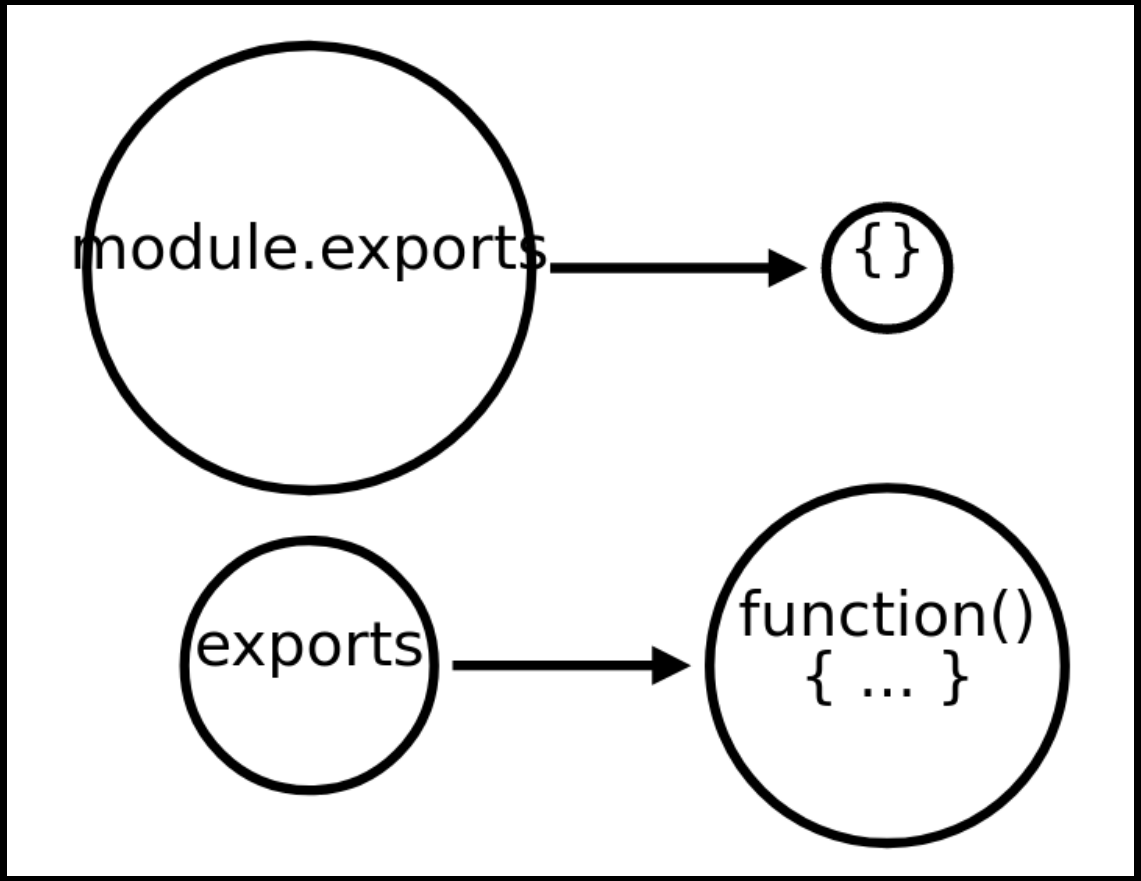
```
// module1.js

exports.a = function() {
    console.log("a");
}
exports.b = function() {
    console.log("b");
}

exports = function extra() {
    console.log('Extra function');
}
```

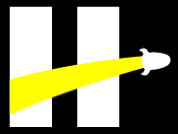
```
// module2.js

var example = require('./module1.js');

example.a(); // "a"

example.b(); // "b"

example.extra() // TypeError: example.extra is not a function

example(); // TypeError: example is not a function
```



module.exports vs exports

Si "pisamos" la referencia del exports...

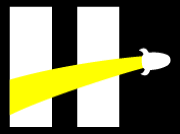




ES6

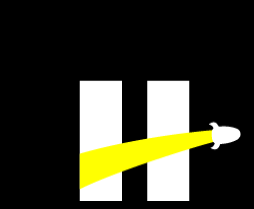
```
// WeekDays.js

var names = ["Domingo", "Lunes", "Martes", "Miercoles",
                "Jueves", "Viernes", "Sabado"];

export function name (number) { return names[number]; };
export function number(name) { return names.indexOf(name); };

export default function myDefault () {
  // otras cosas...
}
```

```
// index.js

import { number, name } from './WeekDays.js';

console.log(name(number("Domingo")));
```

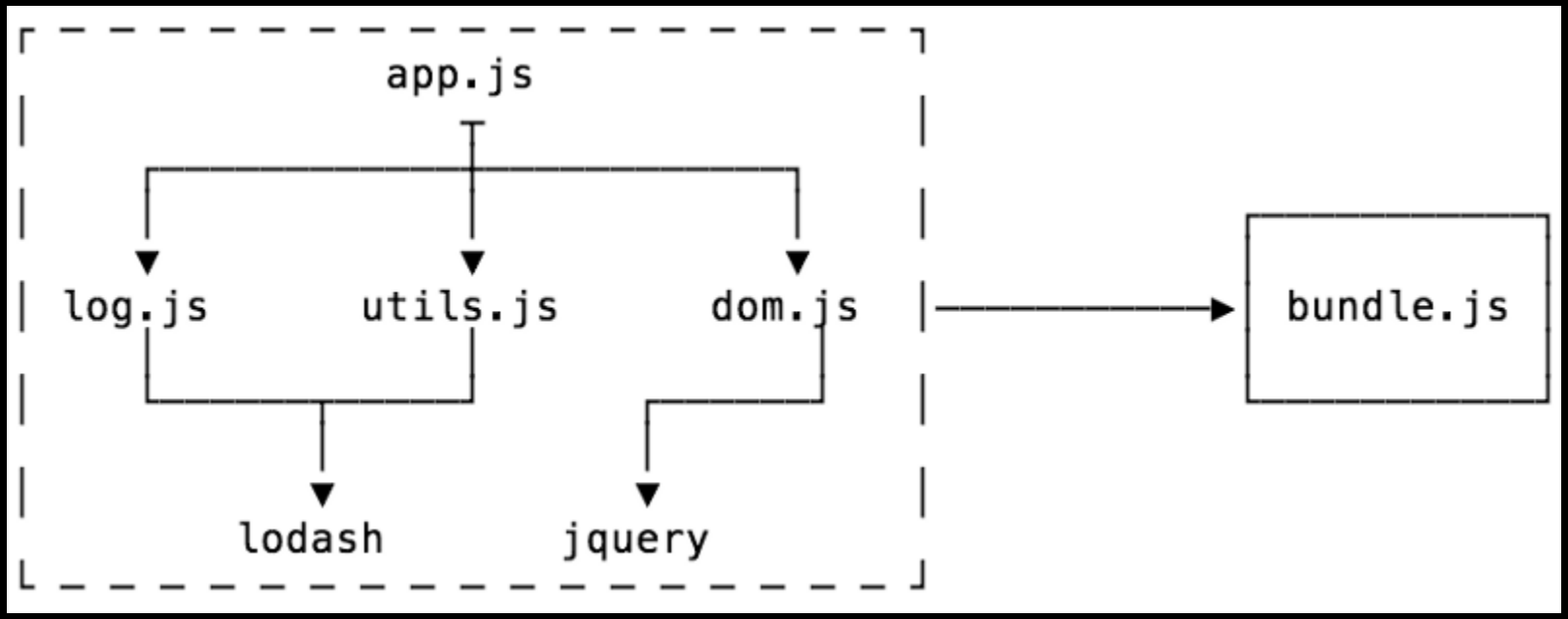


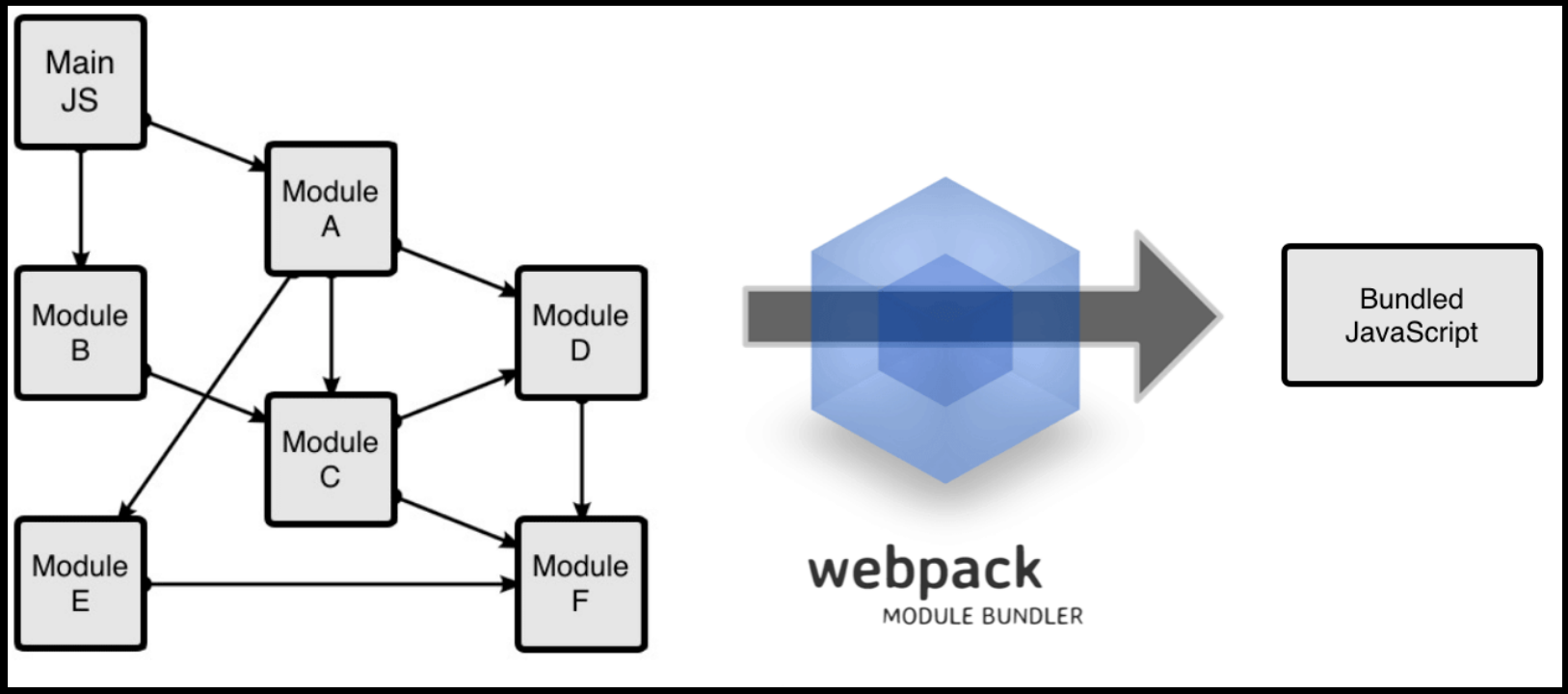
Bundlers

```
<html>
    <head>My Webpage</head>
    <body>
        <script src="script1.js"></script>
        <script src="script2.js"></script>
        <script src="script3.js"></script>
        <script src="script4.js"></script>
    </body>
</html>
```

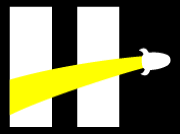


Bundlers



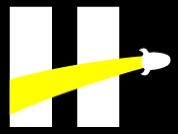

Main JS
Module A
Module B
Module D
Module C
Module E
Module F
webpack
MODULE BUNDLER
Bundled JavaScript

Webpack



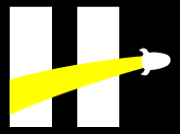
Webpack

```
npm init

npm install --save-dev webpack webpack-cli

// package.json

"scripts": {
  "start": "node server.js",
  "build": "webpack"
}

// crear archivo webpack.config.js`
```



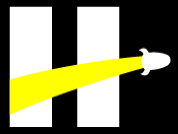
Webpack

```
// webpack.config.js`

module.exports = {
  entry: './browser/app.js', // el punto de arranque de nuestro programa
  output: {
    path: __dirname + '/browser', // el path absoluto para
        // el directorio donde queremos que el output sea colocado
    filename: 'bundle.js' // el nombre del archivo que va a contener
        //nuestro output - podemos nombrarlo como queramos pero bundle.js es lo típico
  }
}
```



Webpack

```
// npm run build

Hash: 43fddf2175fdf6f7923f
Version: webpack 2.2.1
Time: 72ms
    Asset     Size  Chunks             Chunk Names
bundle.js  3.1 kB       0  [emitted]  main
   [0] ./browser/app.js 586 bytes {0} [built]
```



Webpack

```
<!doctype html>
<html>
    <head>
    </head>
    <body>

        <script src="./dist/bundle.js"></script>
    </body>
</html>

```

Incluyendo el script en el HTML