



NodeJS

Un entorno de ejecución de JavaScript orientado a eventos asincrónicos.

Parte I



¿Qué es NodeJS?

- ▶ Node.js es framework Open Source que corre en el servidor.
- ▶ Node.js es libre y gratuito
- ▶ Node.js corre en varias plataformas (Windows, Linux, Unix, Mac OS X, etc.)
- ▶ Node.js usa JavaScript en el servidor
- ▶ Node.js usa programación asincrónica



¿Qué puede hacer con NodeJS?

- ▶ Node.js puede generar contenido dinámico.
- ▶ Node.js puede crear, abrir, leer, escribir, borrar, y cerrar archivos en el servidor.
- ▶ Node.js puede recoger información de formularios.
- ▶ Node.js puede agregar, borrar y modificar datos en una base de datos.



Un archivo NodeJS

- ▶ Un archivo Node.js contiene tareas que serán ejecutadas como respuesta a ciertos eventos.
- ▶ Un evento típico es algo que intent accede a un Puerto del servidor.
- ▶ El archivo Node.js debe ser iniciado en el servidor antes de tener alguna acción.
- ▶ Los archivos Node.js tienen extensión ".js"



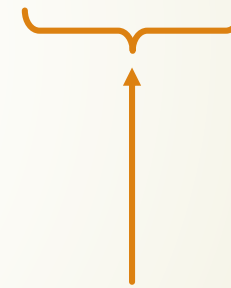
Módulos NodeJS

- ▶ Los módulos son el equivalente a las librerías JavaScript.
- ▶ Un conjunto de funciones que deseamos incluir en nuestra aplicación.
- ▶ Los **Built-in Modules**, son módulos que ya vienen integrados por defecto en Node, por lo que podemos utilizarlos sin tener que instalarlos.

Incluir módulos

Para incorporar módulos, se utiliza el método:

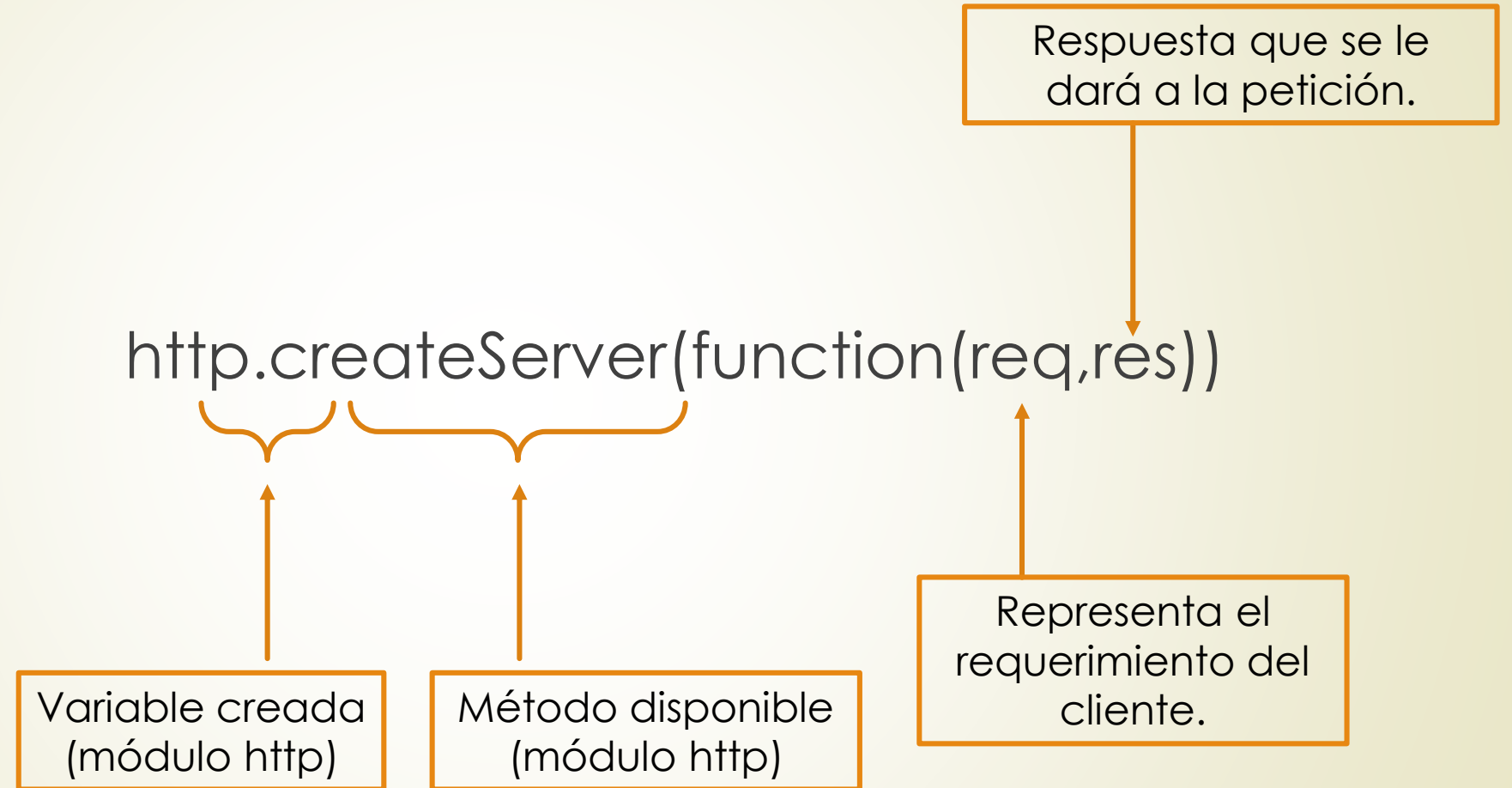
```
var http = require('http')
```



módulo

De esta forma la aplicación tiene acceso al módulo **http** y crear el servidor.

Componentes y métodos



Componentes y métodos

```
res.write(req.url);  
res.end();
```

El objeto tiene una propiedad denominada "**url**" la cuál contiene la parte de la url posterior al nombre de dominio (parámetros)

**res es el objeto
respuesta que
luego
revisamos**



HOLA MUNDO

The diagram shows a Node.js server code snippet with several annotations. An orange arrow points from the 'http' module in the first line to a box labeled 'Carga módulo http.'. Another orange arrow points from the 'http.createServer()' function call to a box labeled 'Creo el servidor.'. A third orange arrow points from the '8080' port in the listen method to a box labeled 'Puerto en el escucho peticiones.'. A fourth orange arrow points from the 'res.writeHead()' method to a box labeled 'Genero respuesta al request.'. A fifth orange arrow points from the 'res.writeHead()' method to a box labeled 'Uso writeHead() si el resultado se muestra en HTML.'. The code is as follows:

```
var http = require('http');  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write('Hello World!');  
  res.end();  
}).listen(8080);
```

Carga módulo http.

```
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write('Hello World!');  
  res.end();  
}).listen(8080);
```

Creo el servidor.

Genero respuesta al request.

Puerto en el escucho peticiones.

Uso writeHead() si el resultado se muestra en HTML.

Parseo de parámetros de url

```
var http = require('http');
```

```
var url = require('url');
```

```
http.createServer(function (req, res) {
```

```
  res.writeHead(200, {'Content-Type': 'text/html'});
```

```
var q = url.parse(req.url, true).query;
```

```
var txt = q.year + " " + q.month;
```

```
res.end(txt);
```

```
}).listen(8080);
```

Concatenación

Parseo de parámetros

```
var url = require('url');
```

```
var adr = 'http://localhost:8080/default.htm?year=2017&month=mayo';
```

```
var q = url.parse(adr, true);
```

```
console.log(q.host); ← retorna 'localhost:8080'
```

```
console.log(q.pathname); ← retorna '/default.htm'
```

```
console.log(q.search); ← 'year=2017&month=mayo'
```

```
var qdata = q.query; ← retorna un objeto: { year: 2017, month: 'mayo' }
```

```
console.log(qdata.month); ← retorna 'mayo'
```