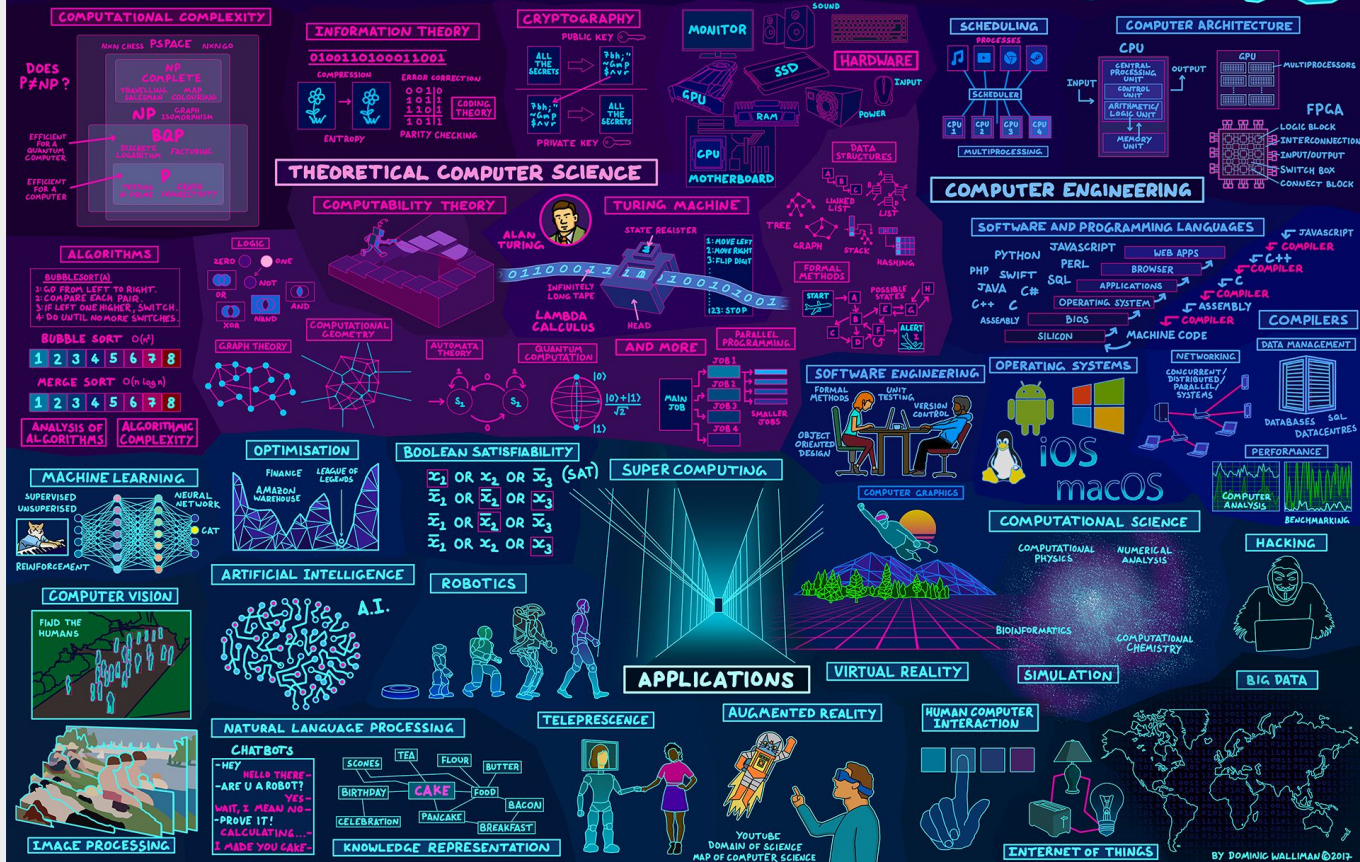




Nuestro compromiso es con el *futuro*.

Introducción a la programación

MAP OF COMPUTER SCIENCE



Clase 1

¿Qué es la programación?



La programación es el proceso utilizado para idear y ordenar las acciones necesarias para realizar un fin determinado, dar instrucciones a algún dispositivo o crear programas para su uso en computadoras.

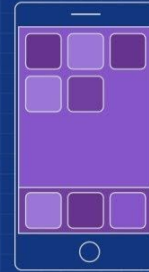
Hoy en día asociamos el término principalmente con la creación de software y aplicaciones de informática.

Software vs Hardware

HARDWARE



SOFTWARE



Software y Sistema Operativo

El **software** es el conjunto de componentes lógicos (código) que hacen posible la realización de tareas específicas, en contraposición de los componentes físicos que son llamados **hardware**.

El **software** incluye entonces, por ejemplo, a distintas aplicaciones informáticas como el software de sistema, videojuegos, aplicaciones de interacción, procesadores de texto, programas de ejecución, etc.

Cuando mencionamos al software de sistema, no podemos dejar de referirnos al que tal vez sea el más importante: el sistema operativo.

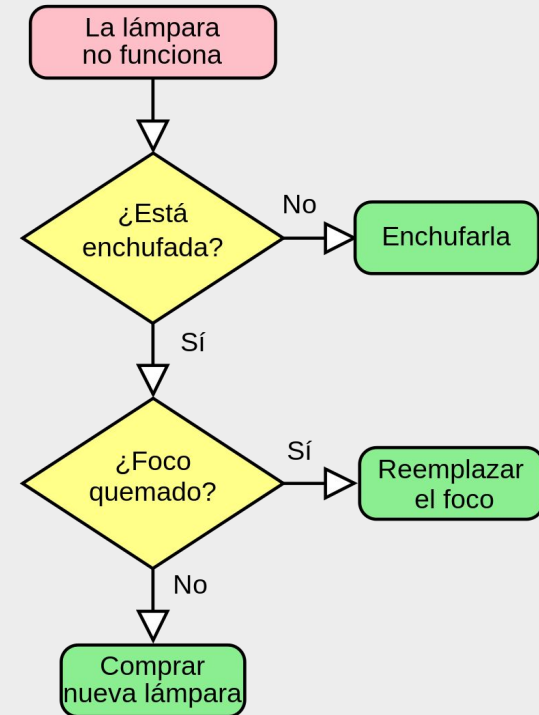
Es este quien gestiona todos los recursos del hardware y provee servicios a los programas de aplicación de software.

Algoritmos

Los algoritmos son un conjunto de instrucciones o reglas **definidas, no-ambiguas, ordenadas y finitas** que permiten solucionar un problema, realizar un cómputo, procesar algún dato o llevar a cabo una tarea específica.

En la vida cotidiana utilizamos todo el tiempo algoritmos para resolver problemas. Cada vez que realizamos una secuencia predefinida de pasos para un fin estamos ejecutando un algoritmo.

Esto puede ser algo tan sencillo como una rutina o una receta, o algo tan complejo como una sucesión de instrucciones técnicas en alguna tarea determinada.



Lenguajes de programación

Un lenguaje de programación es un lenguaje **formal** (es decir con reglas gramaticales y semánticas bien establecidas) que le proporciona a un individuo la posibilidad de determinar una serie de instrucciones o secuencias de órdenes en forma de **algoritmos** para poder controlar el comportamiento físico y lógico de un sistema informático.

Ese conjunto de instrucciones es al que eventualmente llamaremos **programa informático** o **aplicación**.

```
(function sayHi() {  
  console.log("Hola Icaro!")  
})();
```

Lenguajes de programación

Lenguajes
de alto
nivel



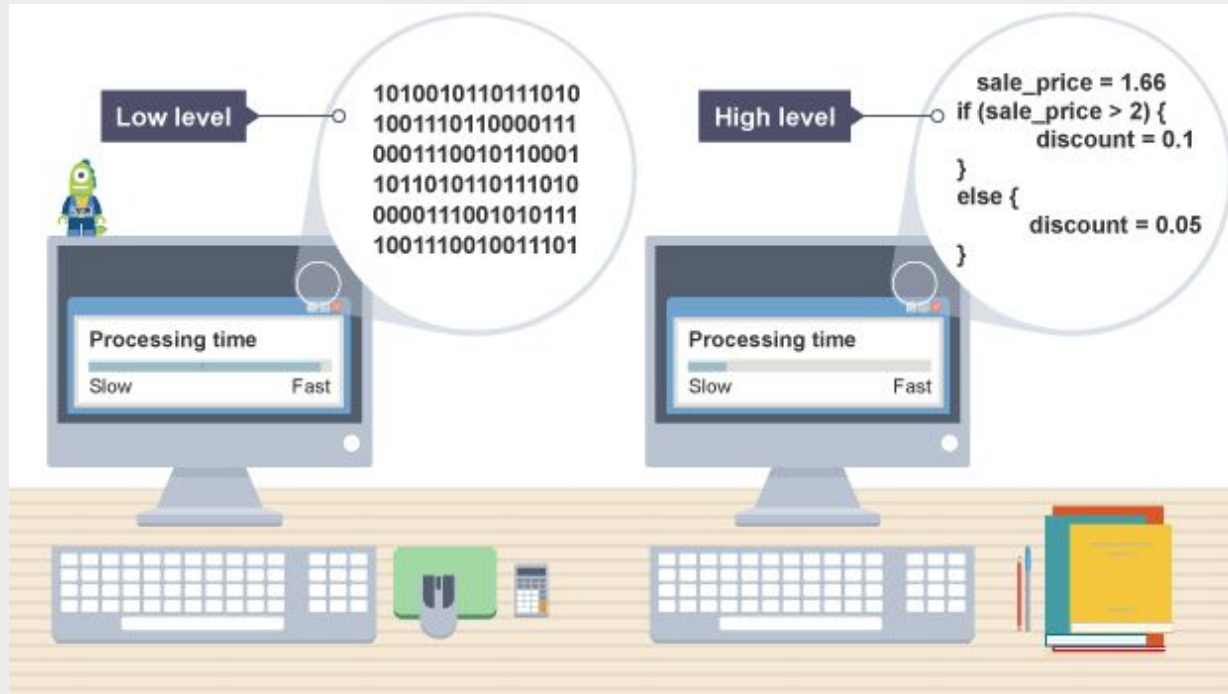
Lenguajes
de bajo
nivel

Lenguajes
interpretados

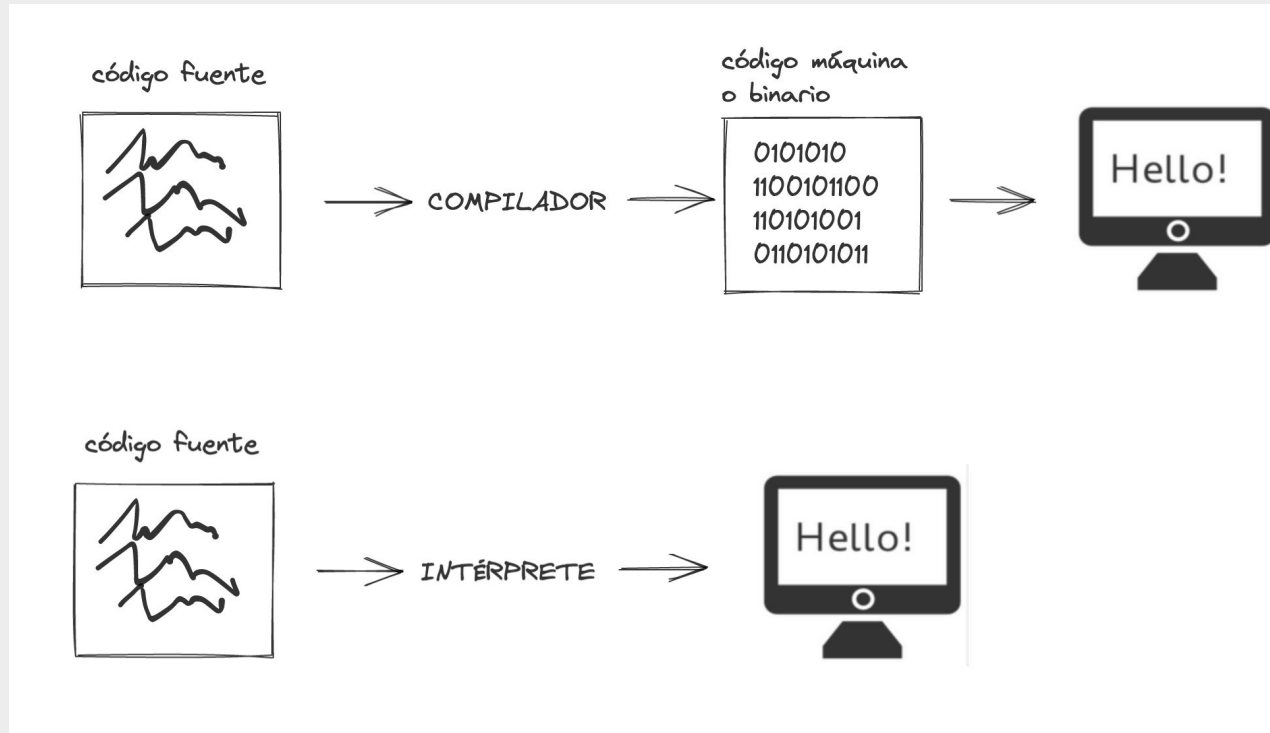


Lenguajes
compilados

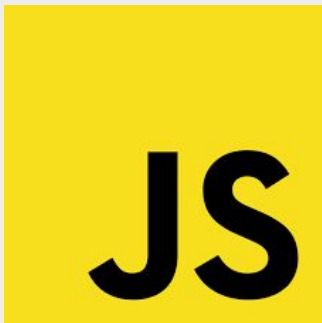
Lenguajes de alto y bajo nivel



Interpretación y compilación

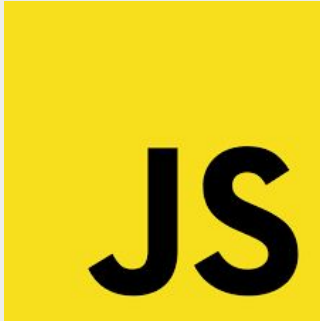


JavaScript



JavaScript (abreviado comúnmente JS) es un lenguaje de programación **interpretado**. Se define como **orientado a objetos**, basado en prototipos, imperativo, **débilmente tipado** y dinámico.

JavaScript



- **interpretado**
- **orientado a objetos**
- **basado en prototipos**
- **imperativo**
- **débilmente tipado**
- **dinámico**

V8



- Es el **motor** de JavaScript en Chrome*
- Es de **código abierto**
- Implementa standard ECMA-262
- Puede ejecutarse **por sí mismo**
- Está escrito en C++

* “¿Alguien conoce ese navegador, Chrome? ¿Es bueno?”

V8 + libuv



- Organiza el código en **módulos**
- Lee y escribe **archivos** de la PC
- Lee y escribe en **bases de datos**
- Envía y recibe **datos por internet**
- Maneja procesos **asincrónicos**

Nodejs

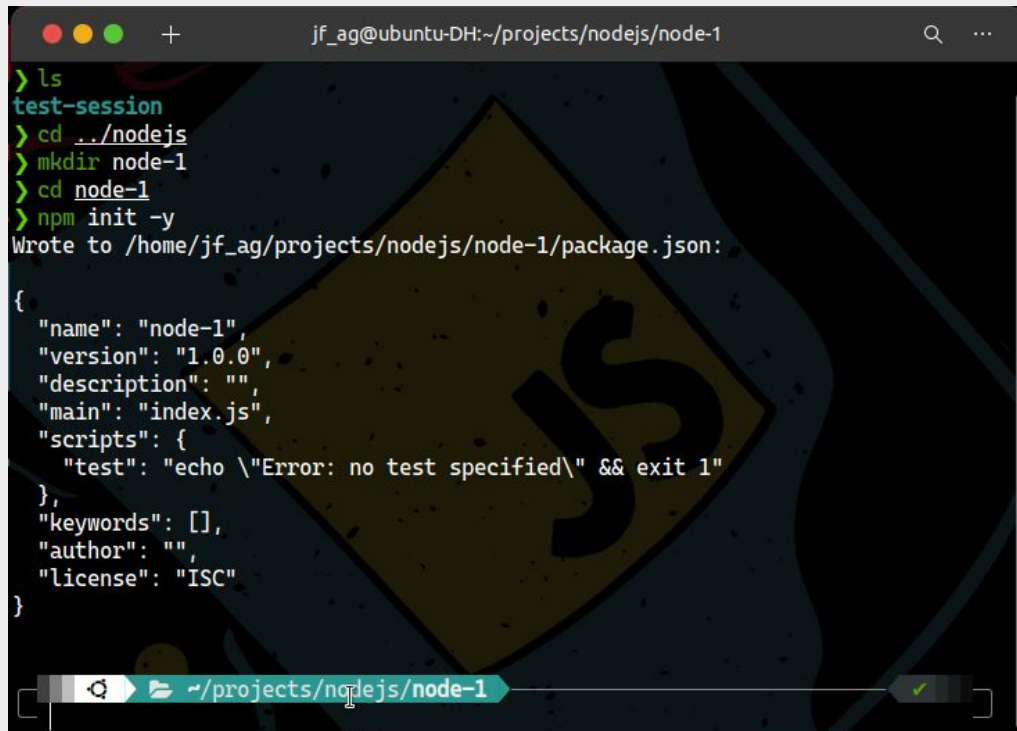


Node.js es un entorno **en tiempo de ejecución** multiplataforma, de código abierto, para la capa del servidor, entre otros usos, basado en el lenguaje de programación JavaScript. Es **asíncrono**, arquitectura orientada a eventos y basado en el **motor V8** de Google.

La consola

La consola, también llamada terminal, es un entorno de trabajo **no gráfico** donde podemos utilizar comandos específicos para indicarle a la computadora qué queremos que haga.

Para comenzar podemos utilizar nuestro **git bash** para probar nuestros primeros comandos.



```

jf_ag@ubuntu-DH:~/projects/nodejs/node-1
> ls
test-session
> cd ../nodejs
> mkdir node-1
> cd node-1
> npm init -y
Wrote to /home/jf_ag/projects/nodejs/node-1/package.json:

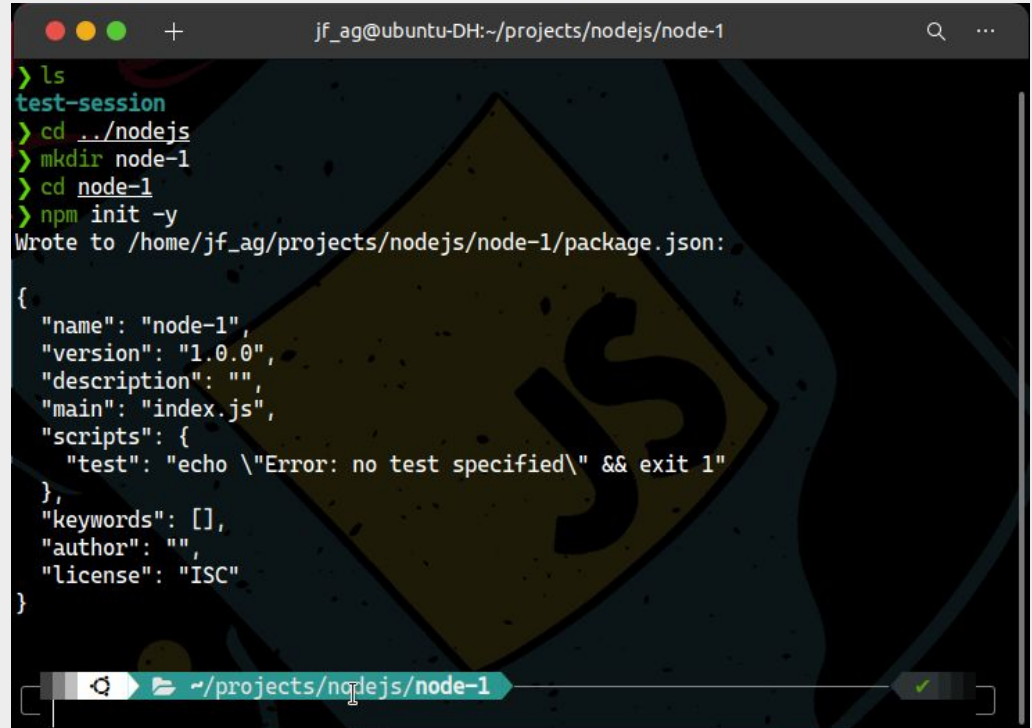
{
  "name": "node-1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
  
```

La consola

REPL

Read
Evaluate
Print
Loop

- leer
- evaluar
- imprimir
- cíclico



```

jf_ag@ubuntu-DH:~/projects/nodejs/node-1
> ls
test-session
> cd ../nodejs
> mkdir node-1
> cd node-1
> npm init -y
Wrote to /home/jf_ag/projects/nodejs/node-1/package.json:

{
  "name": "node-1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
  
```

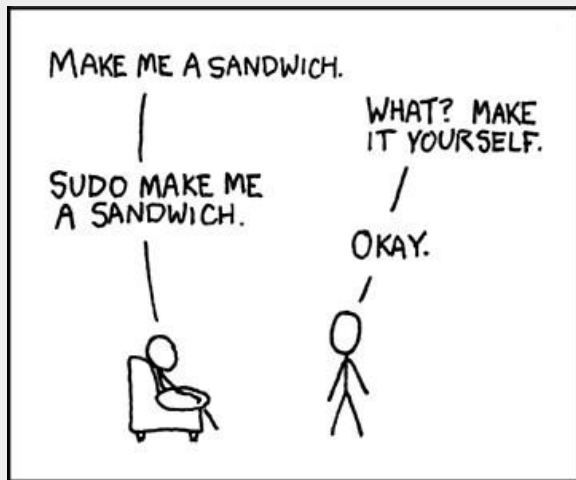
La consola

Cuando estamos dentro de la consola, debemos entender que al no poder utilizar nuestro cursor, todo lo que intentemos realizar deberá ser a través de un input de teclado. Entonces son los **comandos** los que nos permiten realizar las acciones, desde navegar por nuestra computadora, hasta crear archivos, moverlos, modificarlos, borrarlos o incluso reiniciar la computadora, apagarla, activar el bluetooth, realizar una conexión a algún lugar de Internet, etc.

Algo muy importante a tener en cuenta, es que *“en lenguaje consola”* el caracter punto → . ← significa el lugar donde nos encontramos actualmente, es decir la ubicación donde se encuentra “parada” la consola.

Este concepto va a ser clave para poder navegar utilizando la consola o bien ubicar archivos en nuestra computadora, pasándole las rutas necesarias utilizando el caracter barra → / ← para ir navegando entre carpetas o ubicaciones. Veremos más al respecto en la parte práctica.

Comandos comunes



- `cd` → **desplaza** entre directorios.
- `mkdir` → **crea** un nuevo directorio.
- `ls` → **lista** el contenido del directorio.
- `cat` → **muestra** el contenido de un archivo.
- `cp` → **copia** el contenido de un archivo o directorio.
- `mv` → **mueve** o renombra archivos o directorios.
- `sudo` → otorga permisos de **superusuario**

Datos

Antes de comprender cómo funcionan los lenguajes de programación, debemos entender con qué es con lo que trabajan. Esto se hace con **datos**.

Entonces surge la pregunta básica: ¿Qué es un dato?

Un dato es una pieza de información concreta, una cifra, una letra o una palabra que puede ser almacenada o procesada para algún fin. Esto lo podemos hacer nosotros, o bien una máquina.

Nosotros intuitivamente guardamos el conocimiento para darle un uso inmediato, posterior o para procesarlo y obtener nueva información, pero... una computadora no puede hacer esto.

Entonces ¿Cómo logra la computadora hacerlo?

La respuesta es **manejando esos datos** utilizando algoritmos.



Variables

Entonces... comprendimos que las computadoras deben utilizar los datos para realizar las tareas, pero ¿Cómo utilizan estos datos? ¿Cómo saben qué dato usar y cuándo hacerlo?

Para ello existe en la programación el concepto de **variable**, que es un lugar donde se almacena un dato en concreto, lo representa y permite manejarlo. Estas **variables** tienen nombres que permiten diferenciarlas unas de otras, y estos nombres también son llamados **identificadores**, porque justamente cumplen esa función.

Entonces en una variable podemos almacenar un valor (el **dato**) asignándole un identificador que la represente (el **nombre**). Es muy importante destacar que los nombres de las variables tiene ciertas restricciones.

Datos y variables

1 BIT



unidad mínima de información, que puede tener sólo dos valores: **1** o **0**

1 BYTE



conjunto de 8 bits

Tipos de Variables

Números enteros (**int**)
Números enteros grandes (**long**, **bigint**)
Números decimales (**double**, **float**, etc)
Verdadero o Falso (**boolean**)
Símbolo o Caracter (**char**)
Cadenas de texto (**string**)

En JavaScript tenemos menos tipos de datos

Numéricos (**number**)
Texto (**string**)
Booleano (**boolean**)
Object (comprende varios)
null
undefined

Asignación de variables

Asignación de variables

variable = **valor**

numeroAleatorio = **56**

nombreCompleto = **"Juan Aguirre"**

cuotaAlDia = **true**

Nomenclaturas

camelCase ---> **nuevas** palabras con **mayúsculas**

snake_case ---> **nuevas** palabras separadas con **_**

PascalCase ---> **todas** palabras con **mayúsculas**

kebab-case ---> **nuevas** palabras separadas con **-**

Operadores

suma ---> valor **+** valor

resta ---> valor **-** valor

multiplicación ---> valor ***** valor

división ---> valor **/** valor

¡Vamos al código!

Muchas gracias!



ICARO Asociación Civil
CUIT 30716564815
info@icaro.org.ar
www.icaro.org.ar