

# Conceptos básicos de PHP (I)

Ricardo Pérez López

IES Doñana, curso 2018-19

1. Introducción a PHP
2. Sintaxis básica
3. Funcionamiento del intérprete
4. Intérprete interactivo
5. Variables
6. Tipos básicos de datos
7. Manipulación de datos
8. Constantes
9. Ejercicios

# 1. Introducción a PHP

## 1.1. Página web de PHP

`http://php.net`



## 1.2. Instalación de PHP

- Opción recomendada:  
`$ ~/.conf/scripts/php-install.sh`
- Opción general:  
`$ sudo apt install php`

```
// Useless comment.  
alert('hi');
```

When this fragment is shown, the first line of code (`span.line`) will have the `"focus"` class added to it.

Another fragment. This time, both lines will now have the `"focus"` class.

## 1.3. Documentación y búsqueda de información

- Manual en <http://php.net/manual/es/>
- Formulario de búsqueda en <http://php.net>



## 2. Sintaxis básica

## 2.2. Sentencias y comandos

Comando `echo`.

## 2.3. Expresiones, operadores y funciones

Ejemplos: aritmética, `cos()`, `max()`.

### 3. Funcionamiento del intérprete

## 3.1. Modo dual de operación

Se llaman *modo HTML* y *modo PHP*.

## 4. Intérprete interactivo

## 5. Variables

## 5.3. Operadores de asignación

- Por valor
- Por referencia



## 6. Tipos básicos de datos

## 6.1. Lógicos

- Operadores lógicos

## 6.2. Numéricos

- Enteros (`int`)
- Números en coma flotante (`float`)
- Operadores
  - Aritméticos
  - De incremento/decremento

## 6.3. Cadenas

- Operadores de cadenas
  - Concatenación
  - Acceso y modificación por caracteres
  - De incremento/decremento
- Funciones de manejo de cadenas
- Extensión *mbstring*

## 7. Manipulación de datos

## 7.3. Comprobaciones

- De tipos
  - `gettype()`
  - `is_*()` (poco útiles en formularios, ya que sólo se reciben `strings`)
- De valores
  - `is_numeric()`
  - `ctype_*()`

## 7.4. Conversiones de tipos

- Conversión explícita (forzado o *casting*) vs. automática
  - A `bool`
  - A `int`
  - A `float`
  - De `string` a número
  - A `string`
- Funciones de obtención de valores (Hacen más o menos lo mismo que los *casting* pero con funciones en lugar de con operadores. Puede ser interesante porque las funciones se pueden guardar, usar con *map*, *reduce*, etc.)
  - `intval()`
  - `floatval()`
  - `strval()`
  - `boolval()`
- Funciones de formateado numérico
  - `number_format()`
  - `money_format()`
    - `set_locale()`

## 7.5. Comparaciones

- Operadores de comparación
- `==` vs. `===`
- Operador ternario (`? :`)
- Operador de coalescencia o fusión de null (`??`)
- Reglas de comparación de tipos



## 8. Constantes

## 9. Ejercicios