php Cheat Sheet

DEVELOTECA

Sintaxis Básica

- **Variables**: `$variable_name = value;`
- **Comentarios de una línea**: `// Este es un comentario`
- **Comentarios multilínea**: `/* Este es un comentario multilínea */`
- **Declaración de tipo estricto**: `declare(strict_types=1);`
- **Imprimir salida**: `echo "Hola Mundo!";` o `print "Hola Mundo!";`

Tipos de Datos

- **Entero**: `$x = 10;`
- **Flotante**: `$pi = 3.14;`
- **String**: `$name = "John";`
- **Booleano**: `$is_true = true;`
- **Array**: `$colors = array("red", "green", "blue");` o `$colors = ["red", "green", "blue"];` (a partir de PHP 5.4)
- **Objeto**: `$obj = new MyClass();`
- **NULL**: `$x = null;`
- **Resource**: (Referencia externa, ej. conexión a base de datos)

Operadores

- **Aritméticos**: `+`, `-`, `*`, `/`, `%`, `**` (potencia)
- **Asignación**: `=`, `+=`, `-=`, `*=`, `/=`, `%=`
- **Comparación**: `==`, `===` (idéntico), `!=`, `!==` (no idéntico), `>`, `<`, `>=`, `<=`
- **Incremento/Decremento**: `++$x` (pre-incremento), `$x++` (post-incremento), `--$x` (pre-decremento), `$x--` (post-decremento)
- **Lógicos**: `&&` (AND), `||` (OR), `!` (NOT)
- **Concatenación**: `.` (String)
- **Array**: `+` (unión), `==` (igualdad), `===` (identidad), `!=` (desigualdad), `<>` (desigualdad), `!==` (no identidad)

Estructuras de Control

Uso de if

```
if (condicion) {
  // código a ejecutar si la condición es verdadera
} elseif (otra_condicion) {
  // código a ejecutar si la otra condición es verdadera
} else {
  // código a ejecutar si ninguna condición es verdadera
}
```

Uso de switch

```
switch (variable) {
  case valor1:
    // código a ejecutar si variable == valor1
    break;
  case valor2:
    // código a ejecutar si variable == valor2
    break;
  default:
    // código a ejecutar si variable no coincide con ningún caso
}
```

Uso de while

```
while (condicion) {
  // código a ejecutar mientras la condición sea verdadera
}
```

Uso de do...while

```
do {
  // código a ejecutar al menos una vez
} while (condicion);
```

Uso de for

```
for ($i = 0; $i < 10; $i++) {
  // código a ejecutar 10 veces
}
```

Uso de foreach

```
$colors = array("red", "green", "blue");
foreach ($colors as $color) {
  echo $color;
}


$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
foreach($age as $x => $x_value) {
  echo "Key=" . $x . ", Value=" . $x_value;
  echo "";
}
```

Namespaces

```
namespace MyProject\Sub\Level;

class MyClass { ... }

function my_function() { ... }

const MY_CONST = 1;

// Usar el namespace
$obj = new \MyProject\Sub\Level\MyClass();


//Usando 'use'
namespace Foo;
use Bar\Baz\Qux;

$obj = new Qux(); // $obj is an instance of Bar\Baz\Qux
```

Arrays

- **Crear un array**: `$colors = array("red", "green", "blue");` o `$colors = ["red", "green", "blue"];`
- **Acceder a un elemento**: `$first_color = $colors[0];`
- **Añadir un elemento**: `$colors[] = "yellow";`
- **Contar elementos**: `$count = count($colors);`
- **Array asociativo**: `$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");`
- **Acceder array asociativo**: `echo "Peter is " . $age['Peter'] . " years old.";`

Superglobales

- **`$_GET`**: Variables GET (en la URL).
- **`$_POST`**: Variables POST (formulario).
- **`$_COOKIE`**: Cookies.
- **`$_SESSION`**: Sesiones. Requiere `session_start();`
- **`$_SERVER`**: Información del servidor.
- **`$_REQUEST`**: Variables GET, POST y COOKIE.
- **`$_FILES`**: Archivos subidos.
- **`$_ENV`**: Variables de entorno.

Funciones

Definición de una función

```
function myFunction($arg1, $arg2) {
  // código de la función
  return $result;
}
```

Llamada a una función

```
$result = myFunction("value1", "value2");
```

Funciones Anónimas (Closure)

```
$greet = function($name) {
        printf("Hello %s\r\n", $name);
    };

    $greet('World');
```

Manejo de Archivos

Leer un archivo

```
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("webdictionary.txt"));
fclose($myfile);
```

Escribir en un archivo

```
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
```

Seguridad

- **Sanitización de datos**: Usar `htmlspecialchars()`, `strip_tags()`.
- **Validación de datos**: Usar `filter_var()` con filtros como `FILTER_VALIDATE_EMAIL`, `FILTER_VALIDATE_INT`.
- **Prepared statements**: Para prevenir inyección SQL.
- **Evitar mostrar errores en producción**: Configurar `display_errors = Off` en php.ini.

Clases y Objetos

```
class MyClass {
  public $property; // Propiedad pública
  private $private_property; // Propiedad privada
  protected $protected_property; // Propiedad protegida

  public function __construct($value) { // Constructor
    $this->property = $value;
  }

  public function myMethod() { // Método público
    return "Valor: " . $this->property;
  }

  private function myPrivateMethod() { // Método privado
    // Solo accesible dentro de la clase
  }
}

$obj = new MyClass("Ejemplo");
echo $obj->myMethod();
```

Herencia

```
class ChildClass extends MyClass {
    public function childMethod() {
        return "Child: " . $this->property;
    }
}

$childObj = new ChildClass("Heredado");
echo $childObj->childMethod();
```

DEVELOTECA