

Запись на курсы по HTML, CSS, JavaScript, PHP, фреймворкам и CMS,  
**а также:** помощь в поиске работы и заказов, стажировка на реальных проектах→

Верстка JavaScript PHP NodeJs Vue React Laravel WordPress AJAX Парсинг

Бесплатные курсы по React для новичков. Начало 4-го ноября→    Конкурс CSS картинок. Тема: Хэллоун. Призовой фонд: 100\$. Подробности→

# Функции для работы с классами и объектами

## Функция `get_class`

Изучите теорию по следующим ссылкам:  
[get\\_class](#)

### Задача 33.1

Сделайте объект какого-нибудь класса. Примените к объекту функцию **`get_class`** и узнайте имя класса, которому принадлежит объект.

### Задача 33.2

Сделайте два класса: **`Test1`** и **`Test2`**. Пусть оба класса имеют свойство **`name`**.

Создайте некоторое количество объектов этих классов и запишите в массив **`$arr`** в произвольном порядке.

Переберите этот массив циклом и для каждого объекта выведите значение его свойства **`name`** и имя класса, которому принадлежит объект.

## Функция `get_class_methods`

Изучите теорию по следующим ссылкам:  
[get\\_class\\_methods](#)

### Задача 33.3

Сделайте класс **`Test`** с методами **`method1`**, **`method2`** и **`method3`**. С помощью функции **`get_class_methods`** получите массив названий методов класса **`Test`**.

### Задача 33.4

Создайте объект класса **`Test`**, запишите его в переменную **`$test`**.

С помощью функции **`get_class_methods`** получите массив названий методов объекта. Переберите его циклом и в цикле вызовите каждый метод класса, используя объект **`$test`**.

Переберите этот массив циклом и в этом цикле вызовите каждый метод объекта.



# Функция `get_class_vars`

Изучите теорию по следующим ссылкам:

[get\\_class\\_vars](#)

## Задача 33.5

Сделайте класс **Test** с публичными свойствами **prop1** и **prop2**, а также с приватными свойствами **prop3** и **prop4**.

## Задача 33.6

Вызовите функцию **get\_class\_vars** снаружи класса **Test**. Выведите массив доступных свойств.

## Задача 33.7

Вызовите функцию **get\_class\_vars** внутри класса **Test** (например, в конструкторе). Выведите массив доступных свойств.

# Функция `get_object_vars`

Изучите теорию по следующим ссылкам:

[get\\_object\\_vars](#)

## Задача 33.8

Сделайте класс **Test** с публичными свойствами **prop1** и **prop2**, а также с приватными свойствами **prop3** и **prop4**. Создайте объект этого класса.

С помощью функции **get\_object\_vars** получите массив свойств созданного объекта.

# Функция `class_exists`

Изучите теорию по следующим ссылкам:

[class\\_exists](#)

## Задача 33.9

Пусть у вас есть класс **Test1** и нет класса **Test2**.



Проверьте, что выведет функция **class\_exists** для класса **Test1** и для класса **Test2**.

## Задача 33.10

Пусть **GET** параметром в адресную строку передается название класса.

Проверьте, существует ли такой класс. Выведите соответствующее сообщение на экран.

# Функция `method_exists`

Изучите теорию по следующим ссылкам:

[method\\_exists](#)

## Задача 33.11

Сделайте класс **Test** с методом **method1** и без метода **method2**.

Проверьте, что выведет функция **method\_exists** для метода **method1** и для метода **method2**.

## Задача 33.12

Пусть **GET** параметрами в адресную строку передаются название класса и его метод.

Проверьте, существует ли такой класс. Если существует - проверьте существование переданного метода. Если и метод существует - создайте объект данного класса, вызовите указанный метод и выведите результат его работы на экран.

# Функция `property_exists`

Изучите теорию по следующим ссылкам:

[property\\_exists](#)

## Задача 33.13

Сделайте класс **Test** со свойством **prop1** и без свойства **prop2**.

Проверьте, что выведет функция **property\_exists** для свойства **prop1** и для свойства **prop2**.

## Задача 33.14

Дан массив со свойствами класса. Дан также класс, имеющий часть из этих свойств. Переберите этот массив циклом, для каждого свойства проверьте, существует ли оно в классе и, если существует, выведите на экран значение этого свойства.



## Функция `get_parent_class`

Изучите теорию по следующим ссылкам:

[get\\_parent\\_class](#)

### Задача 33.15

Сделайте класс **ChildClass** наследующий от **ParentClass**.

С помощью функции **get\_parent\_class** выведите на экран родителя класса **ParentClass**.

## Функции `is_subclass_of`

Изучите теорию по следующим ссылкам:

[is\\_subclass\\_of](#)

### Задача 33.16

Сделайте класс **ChildClass** наследующий от **ParentClass**, который в свою очередь наследует от **GrandParentClass**

### Задача 33.17

С помощью функции **is\_subclass\_of** проверьте, является ли класс **ChildClass** потомком **GrandParentClass**.

### Задача 33.18

С помощью функции **is\_subclass\_of** проверьте, является ли класс **ParentClass** потомком **GrandParentClass**.

### Задача 33.19

С помощью функции **is\_subclass\_of** проверьте, является ли класс **ChildClass** потомком **ParentClass**.

## Функция `is_a`

Изучите теорию по следующим ссылкам:



[is\\_a](#)

### Задача 33.20

Сделайте класс **ChildClass** наследующий от **ParentClass**. Создайте объект класса **ChildClass**, запишите его в переменную **\$obj**.

### Задача 33.21

С помощью функции **is\_a** проверьте, принадлежит ли объект **\$obj** классу **ChildClass**.

### Задача 33.22

С помощью функции **is\_a** проверьте, принадлежит ли объект **\$obj** классу **ParentClass**.

## Функция `get_declared_classes`

Изучите теорию по следующим ссылкам:  
[get\\_declared\\_classes](#)

### Задача 33.23

Выведите на экран список всех объявленных классов.

