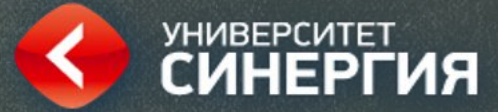


# JSON В PHP

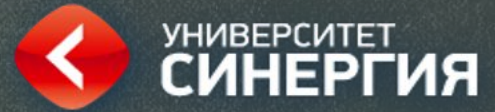
Преподаватель – Борисова Татьяна Андреевна

# Сегодня на уроке:



- Поговорим про JSON в PHP
- Рассмотрим примеры
- Обговорим особенности
- Обсудим домашнее задание

# Что такое JSON



- JSON расшифровывается как аббревиатура от **J**ava **S**cript **O**bject **N**otation. JSON — это стандартный текстовый формат обмена данными, основанный на JavaScript, который представляет собой набор пар {ключ: значение}.
- В JSON, ключи всегда строки, в то время как значение может быть **string**, **number**, **true** или **false**, **null**, **object** или **array**. Строки должны быть заключены в двойные кавычки "и могут содержать управляющие символы, такие как \n, \t и \. JSON более компактный, чем xml, поэтому он более предпочтителен для обмена данными между веб-браузером и сервером.



Объект JSON может выглядеть так:

```
{  
  "ФИО" : "Иванов Иван",  
  "Адрес" : {  
    "Город" : "Киров",  
    "Улица" : "Вербицкого",  
    "Дом" : "7"  
  }  
}
```

В то время как пример массива JSON будет выглядеть примерно так:

```
{  
  "smarts": [  
    "Apple",  
    "ASUS",  
    "Huawei",  
    "Lenovo" ]  
}
```

Структуры данных JSON очень похожи на массивы PHP. В языке PHP существуют специальные функции для кодирования и декодирования данных JSON.

Это функции `json_encode()` и `json_decode()` соответственно. Они уже встроены в ядро PHP, поэтому никаких библиотек подключать не нужно. Обе функции работают только со строковыми данными в кодировке UTF-8.

В PHP функция `json_encode()` используется для кодирования значения в формат JSON. Кодируемое значение может быть любым типом данных PHP, кроме ресурса, такого как база данных или дескриптор файла.

Синтаксис

```
string json_encode ( $value [, int $options = 0 [, int $depth = 512 ]] )
```

Value- значение, которое будет закодировано. Может быть любого типа за исключением resource. Все строковые данные должны быть в кодировке UTF-8.

Options - Битовая маска составляемая из значений JSON\_HEX\_QUOT, JSON\_HEX\_TAG, JSON\_HEX\_AMP, JSON\_HEX\_APOS, JSON\_NUMERIC\_CHECK, JSON\_PRETTY\_PRINT, JSON\_UNESCAPED\_SLASHES, JSON\_FORCE\_OBJECT, JSON\_UNESCAPED\_UNICODE.

Depth - Задаёт максимальную глубину. Должен быть больше нуля.



# Пример

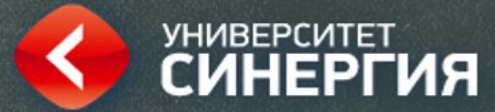
В приведенном ниже примере показано, как кодировать ассоциативный массив в объект JSON:

```
<?php  
  
// ассоциативный массив  
  
$marks = array("Peter"=>65, "Harry"=>80, "John"=>78, "Clark"=>90);  
  
echo json_encode($marks);  
  
?>
```

Результат приведенного выше примера будет выглядеть так:

```
{"Peter": 65, "Harry": 80, "John": 78, "Clark": 90}
```

# Пример



В следующем примере посмотрим, как будут кодироваться кириллические символы:

```
<?php
// ассоциативный массив
$marks = array("Пётр"=>65, "Игорь"=>80, "Денис"=>78, "Николай"=>90);
echo json_encode($marks);
?>
```

Результат приведенного выше примера будет выглядеть так:

```
{"\u041f\u044d\u0442\u0440":65,"\u0418\u0433\u043e\u0440\u044c":80,"\u0414\u0435\u043d\u0438\u0441":78,"\u041d\u0438\u043a\u043e\u043b\u0430\u0439":90}
```

# Пример

Как видно кириллица кодируется, исправляется это добавлением опции: `JSON_UNESCAPED_UNICODE`.

```
<?php
```

```
$marks = array("Пётр"=>65, "Игорь"=>80, "Денис"=>78, "Николай"=>90);
```

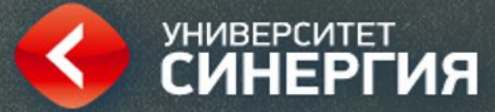
```
echo json_encode($marks, JSON_UNESCAPED_UNICODE);
```

```
?>
```

Результат выполнения кода:

```
{"Пётр": 65, "Игорь": 80, "Денис": 78, "Николай": 90}
```

# Пример



Точно так же вы можете закодировать индексированный массив PHP в массив JSON, например:

В следующем примере посмотрим, как будут кодироваться кириллические символы:

```
<?php
```

```
// индексированный массив
```

```
$colors = array("Красный", "Зелёный", "Синий", "Оранжевый", "Жёлтый");
```

```
echo json_encode($colors, JSON_UNESCAPED_UNICODE);
```

```
?>
```

Результат выполнения кода:

```
["Красный","Зелёный","Синий","Оранжевый","Жёлтый"]
```



# Пример



Если применить параметр `JSON_FORCE_OBJECT`, то функция `json_encode()` будет возвращать индексированный массив PHP как объект JSON:

```
<?php
```

```
// индексированный массив
```

```
$smarts = array("Apple", "ASUS", "Huawei", "Lenovo", "Samsung");
```

```
echo json_encode($smarts, JSON_FORCE_OBJECT);
```

```
?>
```

Результат выполнения кода:

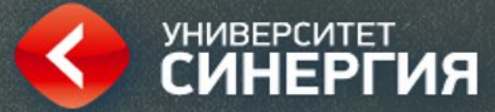
```
{"0":"Apple","1":"ASUS","2":"Huawei","3":"Lenovo","4":"Samsung"}
```

## Примечание



Неассоциативный массив может быть закодирован как массив или объект. Однако ассоциативный массив всегда кодируется как объект.

# Декодирование данных в JSON

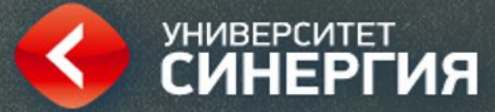


Функция PHP `json_decode()` используется для декодирования JSON в PHP. Эта функция возвращает значение, декодированное из json в соответствующий тип данных PHP.

Синтаксис

```
mixed json_decode ( string $json [, bool $assoc = false [, int $depth = 512 [, int $options = 0 ]]] )
```

# Параметры



json- json строка (string) для декодирования. Эта функция работает только со строками в UTF-8 кодировке.

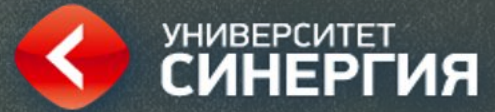
Assoc - Если TRUE, возвращаемые объекты будут преобразованы в ассоциативные массивы.

Depth - Указывает глубину рекурсии.

Options - Битовая маска опций декодирования JSON. В настоящий момент поддерживается только JSON\_BIGINT\_AS\_STRING (по умолчанию большие целые числа приводятся к числам с плавающей запятой (float))



# Пример



В следующем примере показано, как декодировать или преобразовать объект JSON в объект PHP:

```
<?php
```

```
$json = '{"Peter":65,"Harry":80,"John":78,"Clark":90}';
```

```
var_dump(json_decode($json));
```

```
?>
```

Результат выполнения кода:

```
object(stdClass)#1 (4) { ["Peter"]=> int(65) ["Harry"]=> int(80) ["John"]=> int(78) ["Clark"]=> int(90) }
```

# Пример



Функция `json_decode()` по умолчанию возвращает объект. Если нужно JSON декодировать в ассоциативный массив, то укажите значение второго параметра `$assoc` в качестве `true`. По умолчанию его значение — `false`:

```
<?php
```

```
$json = '{"Peter":65,"Harry":80,"John":78,"Clark":90}';
```

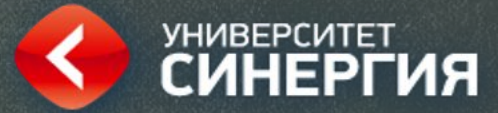
```
var_dump(json_decode($json, true));
```

```
?>
```

Результат выполнения кода:

```
array(4) { ["Peter"]=> int(65) ["Harry"]=> int(80) ["John"]=> int(78) ["Clark"]=> int(90) }
```

# Домашнее задание



## Задание 1

Прочитать дополнительную информацию из официального источника  
<https://www.php.net/manual/ru/book.json.php>

## Задание 2

Составить по пройденному материалу примеры на любую тему

*СПАСИБО ЗА  
ВНИМАНИЕ!*

