

continue ¶

`continue` используется внутри циклических структур для пропуска оставшейся части текущей итерации цикла и, при соблюдении условий, начала следующей итерации.

Давайте возьмём пример

```
for ($i = 1; $i <= 20; $i++){  
    if ($i == 3){  
        continue;  
    }  
    echo "$i <br>";  
}
```

А теперь представим что нам нужно вывести только чётные числа или не четное,

Всё просто нам нужно просто перед выводом информацией проверить чётное число или нечётное с помощью условия,

вспоминаем про оператор деления по модулю
`var_dump(4 % 2);`
если число делится без остатка то число чётное

Давайте выведем чётные числа

```
for ($i = 1; $i <= 20; $i++){  
    if ($i % 2 != 0){  
        continue;  
    }  
}
```

```
echo "$i <br>";  
}
```

`continue` принимает необязательный числовой аргумент, который указывает на скольких уровнях вложенных циклов будет пропущена оставшаяся часть итерации. Значением по умолчанию является 1, при которой пропускается оставшаяся часть текущего цикла.

Давайте посмотрим как это работает

```
for ($i = 1; $i <= 20; $i++){  
  for ($j = 1; $j < 20; $j++) {  
    if ($j == 6) {  
      continue 2;  
    }  
    echo $j . "<br>";  
  }  
}
```

Когда Я указал двойку У нас прерывается не только текущие цикл Ну и основной

break ¶

При каких ситуациях нам понадобится Break например запрос IP нам нужно постоянно что-то запрашивать мы с паузой

```
$i = 1;
```

```
while (true) {  
  echo $i . "<br>";  
  if ($i == 20){  
    echo "число найдено";  
    break;  
  }  
}
```

```
$i++;  
}
```