



### Идентичность

<code>id(obj)</code>	уникальный идентификатор
<code>-6, ..., 256</code>	числа в кэше
<code>'a', ..., 'z'</code> <code>'A', ..., 'Z'</code> и др. символы	кэшируемые строки
<code>a == b</code>	равенство (по значению)
<code>a is b</code>	равенство (по ссылке)

### Множества (set)

<code>{1, 2, 3, 42, 169}</code>	создание явно
<code>set(obj)</code>	создание из итерируемого объекта <i>obj</i>

### Операции над множествами

<code>s.union(other)</code>	объединение
<code>s.intersection(other)</code>	пересечение
<code>s.difference(other)</code>	разность
<code>s.symmetric_difference(other)</code>	симметричная разность

### Логические операторы

<code>and</code>	если все операнды истинные, возвращается последний
<code>and</code>	если один из операндов ложные, возвращается первый ложный
<code>or</code>	если все операнды ложные, то возвращается последний
<code>or</code>	если хотя бы один операнд истинный, то возвращается он, а остальные игнорируются

### Все или любой

<code>any(iter_obj)</code>	возвращается True, если хотя бы один элемент <i>iter_obj</i> истинные
<code>all(iter_obj)</code>	возвращается True тогда и только тогда, когда все элементы <i>iter_obj</i> истинные

### Генератор списка

<code>[i for i in iter_obj]</code>	
	простой генератор
<code>[i for i in iter_obj if cond]</code>	
	генератор с условием

### zip

<code>for a,b in zip(A,B):</code> <code># какой-то код</code>
склеивание двух списков поэлементно

### map

<code>map(func, iter1, iter2, ...)</code>
применяет <b>func</b> к каждому элементу итерируемых объектов
возвращает итератор результатов применения функции

### filter

<code>filter(func, iter1)</code>
функция <b>func</b> возвращает <i>True</i> или <i>False</i>
результат – итератор элементов, для которых <b>func</b> возвращает <i>True</i>

### lambda-функции

<code>lambda x1,x2 : x1+x2</code>
создание анонимной функции
выполняют только одну инструкцию