In [1]:
```
# https://github.com/30-seconds/30_seconds_of_knowledge
```

In [2]:
```
# ---01---
```

In [3]:
```
s = "programming is awesome"
s.title()
```

Out[3]:
```
'Programming Is Awesome'
```

In [4]:
```
# ---02---
```

In [5]:
```
from math import ceil

def chunk(lst, size):
    return list(map(lambda x: lst[x*size:x*size+size], list(range(0,ceil(len(lst)/size)))))
```

In [6]:
```
chunk([1,2,3,4,5], 2)
```

Out[6]:
```
[[1, 2], [3, 4], [5]]
```

In [7]:
```
chunk([1,2,3,4,5], 3)
```

Out[7]:
```
[[1, 2, 3], [4, 5]]
```

In [8]:
```
# ---03---
```

In [9]:
```
array = [["a","b"], ["c","d"], ["e","f"]]
list(zip(*array))
```

Out[9]:
```
[('a', 'c', 'e'), ('b', 'd', 'f')]
```

In [10]:
```
# ---04---
```

In [11]:
```
def add(x1, x2):
    return x1 + x2

def subtract(x1, x2):
    return x1 - x2
```

In [12]:
```
a, b = 4, 5
(subtract if a > b else add)(a, b)
```

Out[12]:
```
9
```

In [13]:
```
a, b = 5, 4
(subtract if a > b else add)(a, b)
```

Out[13]:
```
1
```

In [14]:
```
# ---05---
```

In [15]:
```
def merge_two_dicts(a, b):
    c = a.copy()
    c.update(b)
    return c
a = {"x": 1,"y": 2}
b = {"y": 3,"z": 4}
merge_two_dicts(a, b)
```

Out[15]:
```
{'x': 1, 'y': 3, 'z': 4}
```

In [16]:
```
# ---06---
```

In [17]:
```
def merge_dictionaries(a, b):
   return {**a, **b}
a = {"x": 1,"y": 2}
b = {"y": 3,"z": 4}
merge_dictionaries(a, b)
```

Out[17]:
```
{'x': 1, 'y': 3, 'z': 4}
```

In [18]:
```
# ---07---
```

In [19]:

```
import operator
action = {
    "+": operator.add,
    "-": operator.sub,
    "/": operator.truediv,
    "*": operator.mul,
    "**": pow
}
action["-"](50, 25)
```

Out[19]:

```
25
```

In [20]:

```
# ---08---
```

In [21]:

```
def spread(arg):
    ret = []
    for i in arg:
        if isinstance(i, list):
            ret.extend(i)
        else:
            ret.append(i)
    return ret
x = [1,2,3,[4,5,6],[7],8,9]
x, spread(x)
```

Out[21]:

```
([1, 2, 3, [4, 5, 6], [7], 8, 9], [1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [22]:

```
# ---09---
```

In [23]:

```
d = {"a": 1,"b": 2}
d.get("a"),d.get("a",3), d.get("c"),d.get("c",3)
```

Out[23]:

```
(1, 1, None, 3)
```