

Słowniki

- Słowniki w Pythonie tworzone są z wykorzystaniem nawiasów klamrowych.
- Elementy słowników przechowywane są w postaci par klucz: wartość.
- Klucz pełni funkcję identyfikatora musi być zatem unikalny.

Słowniki

- Słowniki w Pythonie tworzone są z wykorzystaniem nawiasów klamrowych.
- Elementy słowników przechowywane są w postaci par klucz: wartość.
- Klucz pełni funkcję identyfikatora musi być zatem unikalny.
- Klucz musi być obiektem niemodyfikowalnym!

Słowniki

- Słowniki w Pythonie tworzone są z wykorzystaniem nawiasów klamrowych.
- Elementy słowników przechowywane są w postaci par klucz: wartość.
- Klucz pełni funkcję identyfikatora musi być zatem unikalny.
- Klucz musi być obiektem niemodyfikowalnym!

Oznacza to, że kluczem może być literał liczbowy, string lub krotka (ale nie zawierająca obiektów modyfikowanych).

`s={}` **or** `s=collections.defaultdict(int)`

```
s={}      or      s=collections.defaultdict(int)
```

```
print(s[7])                                #KeyError: 7 or 0
```

```
s={}      or      s=collections.defaultdict(int)
```

```
print(s[7])      #KeyError: 7 or 0
```

```
s[7]=3
```

```
print(s)      #{7: 3}
```

```
s={}      or      s=collections.defaultdict(int)
```

```
print(s[7])      #KeyError: 7 or 0
```

```
s[7]=3
```

```
print(s)      #{7: 3}
```

```
s[7]=13
```

```
print(s)      #{7: 13}
```

```
s={}      or      s=collections.defaultdict(int)
```

```
print(s[7])      #KeyError: 7 or 0
```

```
s[7]=3  
print(s)      #{7: 3}
```

```
s[7]=13  
print(s)      #{7: 13}
```

```
w=s.setdefault(7, None)  
print(w)      #13  
print(s)      #{7: 13}
```

```
w=s.setdefault(8, 15)  
print(w)      #15  
print(s)      #{7: 13, 8: 15}
```



```
s={}      or      s=collections.defaultdict(int)
```

```
print(s[7])                                #KeyError: 7 or 0
```

```
s[7]=3  
print(s)                                   #{7: 3}
```

```
s[7]=13  
print(s)                                   #{7: 13}
```

```
w=s.setdefault(7, None)  
print(w)                                   #13  
print(s)                                   #{7: 13}
```

```
w=s.setdefault(8, 15)  
print(w)                                   #15  
print(s)                                   #{7: 13, 8: 15}
```

```
s[9]+=1  
print(s)                                   #KeyError: 9 or {7: 13, 8: 15, 9: 1}
```

Słowniki

```
s={}
s.setdefault(7, []).append(3)
print(s)

s.setdefault(7, []).append(13)
print(s)
```

#{7: [3]}

#{7: [3, 13]}

Słowniki

```
s={}
s.setdefault(7, []).append(3)
print(s)                                #{7: [3]}
```

```
s.setdefault(7, []).append(13)
print(s)                                #{7: [3, 13]}
```

```
s.setdefault(8, {}).setdefault(3, 1)
print(s)                                #{7: [3, 13], 8: {3: 1}}
```

```
s.setdefault(9, {}).setdefault(13, []).append(1)
print(s)                                #{7: [3, 13], 8: {3: 1}, 9: {13: [1]}}
```

Słowniki

```
s=dict.fromkeys(range(5))  
print(s)           #{0: None, 1: None, 2: None, 3: None, 4: None}
```

```
s=dict.fromkeys(range(5), 3)  
print(s)           #{0: 3, 1: 3, 2: 3, 3: 3, 4: 3}
```

Słowniki

```
s=dict.fromkeys(range(5))  
print(s)           #{0: None, 1: None, 2: None, 3: None, 4: None}
```

```
s=dict.fromkeys(range(5), 3)  
print(s)           #{0: 3, 1: 3, 2: 3, 3: 3, 4: 3}
```

```
s=dict((k, f'{k}') for k in range(5))  
print(s)           #{0: '0', 1: '1', 2: '2', 3: '3', 4: '4'}
```

Słowniki

```
s=dict.fromkeys(range(5))  
print(s)           #{0: None, 1: None, 2: None, 3: None, 4: None}
```

```
s=dict.fromkeys(range(5), 3)  
print(s)           #{0: 3, 1: 3, 2: 3, 3: 3, 4: 3}
```

```
s=dict((k, f'{k}') for k in range(5))  
print(s)           #{0: '0', 1: '1', 2: '2', 3: '3', 4: '4'}
```

```
s={k: f'{k}' for k in range(5)}  
print(s)           #{0: '0', 1: '1', 2: '2', 3: '3', 4: '4'}  
print(type(s))     #dict
```

Słowniki

```
s=dict.fromkeys(range(5))  
print(s)           #{0: None, 1: None, 2: None, 3: None, 4: None}
```

```
s=dict.fromkeys(range(5), 3)  
print(s)           #{0: 3, 1: 3, 2: 3, 3: 3, 4: 3}
```

```
s=dict((k, f'{k}') for k in range(5))  
print(s)           #{0: '0', 1: '1', 2: '2', 3: '3', 4: '4'}
```

```
s={k: f'{k}' for k in range(5)}  
print(s)           #{0: '0', 1: '1', 2: '2', 3: '3', 4: '4'}  
print(type(s))     #dict
```

```
###  
s={(k, f'{k}') for k in range(5)}  
print(s)           #{(0, '0'), (1, '1'), (2, '2'), (4, '4'), (3, '3')}  
print(type(s))     #set
```

Słowniki

```
s={7: 13, 8: 15}
```

```
s.keys()
```

```
#dict_keys([7, 8])
```

```
s.values()
```

```
#dict_values([13, 15])
```

```
s.items()
```

```
#dict_items([(7, 13), (8, 15)])
```


Słowniki

```
s={7: 13, 8: 15}
```

```
s.keys()           #dict_keys([7, 8])  
s.values()         #dict_values([13, 15])  
s.items()          #dict_items([(7, 13), (8, 15)])
```

```
print(7 in s)       #True  
print(10 in s)      #False
```

Słowniki

```
s={7: 13, 8: 15}
```

```
s.keys()           #dict_keys([7, 8])  
s.values()         #dict_values([13, 15])  
s.items()          #dict_items([(7, 13), (8, 15)])
```

```
print(7 in s)       #True  
print(10 in s)      #False
```

```
for k in s:  
    print(k, s[k])
```

```
for k,v in s.items():  
    print(k,v)
```

Słowniki

```
s={7: 13, 8: 15}
```

```
del s[10]          #wyjątek
```

Słowniki

```
s={7: 13, 8: 15}
```

```
del s[10]           #wyjątek
```

```
s.get(10)           #None
```

```
s.get(10, 0)        #0
```

Słowniki

```
s={7: 13, 8: 15}
```

```
del s[10]           #wyjątek
```

```
s.get(10)           #None  
s.get(10, 0)        #0
```

```
s.pop(10)           #wyjątek  
s.pop(10, 0)        #0
```

Słowniki

```
s={7: 13, 8: 15}
```

```
del s[10]           #wyjątek
```

```
s.get(10)           #None  
s.get(10, 0)         #0
```

```
s.pop(10)           #wyjątek  
s.pop(10, 0)         #0
```

```
s.clear()
```

Łączenie słowników

```
s1={'k1': 's1', 'k2': 's1', 'k3': 's1', 'k4': 's1'}
s2={'k1': 's2', 'k2': 's2', 'k3': 's2', 'k4': 's2', 'k5': 's2'}
s1.update(s2)
print(s1)          #{'k1': 's2', 'k2': 's2', 'k3': 's2', 'k4': 's2', 'k5': 's2'}
print(s2)          #{'k1': 's2', 'k2': 's2', 'k3': 's2', 'k4': 's2', 'k5': 's2'}
```

```
s1={'k1': 's1', 'k2': 's1', 'k3': 's1', 'k4': 's1'}
s2={'k1': 's2', 'k2': 's2', 'k3': 's2', 'k4': 's2', 'k5': 's2'}
s2.update(s1)
print(s1)          #{'k1': 's1', 'k2': 's1', 'k3': 's1', 'k4': 's1'}
print(s2)          #{'k1': 's1', 'k2': 's1', 'k3': 's1', 'k4': 's1', 'k5': 's2'}
```

>= 3.9

```
s1={'k1': 's1', 'k2': 's1', 'k3': 's1', 'k4': 's1'}
s2={'k1': 's2', 'k2': 's2', 'k3': 's2', 'k4': 's2', 'k5': 's2'}
print(s2|s1)       #{'k1': 's1', 'k2': 's1', 'k3': 's1', 'k4': 's1', 'k5': 's2'}
print(s1)          #{'k1': 's1', 'k2': 's1', 'k3': 's1', 'k4': 's1'}
print(s2)          #{'k1': 's2', 'k2': 's2', 'k3': 's2', 'k4': 's2', 'k5': 's2'}
```

```
s1={'k1': 's1', 'k2': 's1', 'k3': 's1', 'k4': 's1'}
s2={'k1': 's2', 'k2': 's2', 'k3': 's2', 'k4': 's2', 'k5': 's2'}
s2|=s1
print(s1)          #{'k1': 's1', 'k2': 's1', 'k3': 's1', 'k4': 's1'}
print(s2)          #{'k1': 's1', 'k2': 's1', 'k3': 's1', 'k4': 's1', 'k5': 's2'}
```

Liczby pseudolosowe

```
import random
```

```
#liczba całkowita z przedziału [0, koniec)
```

```
random.randrange([początek,] koniec [, krok])
```

```
#liczba całkowita z przedziału [początek, koniec]
```

```
random.randint(początek, koniec)
```

```
#liczba rzeczywista z przedziału [0,1)
```

```
random.random()
```

```
#liczba rzeczywista z przedziału [początek, koniec)
```

```
random.uniform(początek, koniec)
```

```
#betavariate, expovariate, gammavariate, gauss, lognormvariate, normalvariate, paretovariate, triangular,
```

```
#vonmisesvariate, weibullvariate
```


Liczby pseudolosowe

```
import random
```

```
r=random.randbytes(5)
```

```
print(r)
```

```
r=[chr(el) for el in r]
```

```
print(r)
```

```
#print(ord(chr(13)))
```

```
#b'L\xbd\xd6\xb1#'
```

```
#[ 'L', '½', 'Ö', '±', '#']
```

Liczby pseudolosowe

```
import random
```

```
r=random.randbytes(5)
```

```
print(r)
```

```
r=[chr(el) for el in r]
```

```
print(r)
```

```
#print(ord(chr(13)))
```

```
#b'L\xbd\xd6\b1#'
```

```
#[ 'L', '½', 'Ö', '±', '#']
```

```
r=random.getrandbits(5)
```

```
print(r)
```

```
r=bin(r)
```

```
print(r, type(r))
```

```
r=int(r,2)
```

```
print(r)
```

```
#23
```

```
#0b10111, str
```

```
#23
```

Liczby pseudolosowe

```
import random
```

```
#ustawienie ziarna, domyślnie ziarno ustawiane jest na czas systemowy  
random.seed()
```

```
#odpowiednio pobiera i ustawia stan generatora  
random.getstate(), random.setstate()
```

```
#przemieszczenie w miejscu elementów populacji  
random.shuffle(populacja)
```

```
#losuje pojedynczą wartość z populacji  
random.choice(populacja)
```

```
#losuje k różnych elementów z populacji, przy czym  $k \leq \text{len}(\text{populacja})$   
random.sample(populacja, k)
```

```
#od wersji 3.6, losuje k elementów z populacji z powtórzeniami  
random.choices(populacja, k=k)
```