

«Сейфуллин оқулары – 16: Жаңа формациядағы жастар ғылыми – Қазақстанның болашағы» атты халықаралық ғылыми-теориялық конференциясының материалдары = Материалы Международной научно-теоретической конференции «Сейфуллинские чтения – 16: Молодежная наука новой формации – будущее Казахстана». - 2020. - Т.І, Ч.3 - С.151-153

## PYTHON ПРОГРАММАЛАУ ТІЛІНДЕ СӨЗДІК ҚҰРУДЫҢ ӘДІСІ

*Джумағалиева А.М.*

Заман талабына сай көптілді меңгеру қоғам сұранысына сай жеке тұлға қалыптастыру мәселесі бойынша студенттерге ақпараттық - коммуникациялық технологияларды оқу іс-әрекетте қолдану тиімділігі артып отыр. Сөздік - кілт-мән жұптарын сақтауға мүмкіндік беретін реттелмеген деректер құрылымы. Бұл сөздік жолдарды кілт ретінде пайдаланады, бірақ кілт негізінен өзгермейтін деректер түріне айналуы мүмкін. Белгілі бір кілттің мәні кез келген нәрсе болуы мүмкін. Сөздікті Python программалау тілінің амалымен сөздіктерді құру тәсілдерін қарастырамыз: Ондағы берілгендер типтері, кілттер, сөздіктерге қолданылатын мәндер жиынтығын қолданамыз[1,2].

Python-дағы сөздіктер - бұл қол жетімді ерікті нысандардың реттелмеген жиынтығы. Оларды кейде ассоциативті массивтер немесе хэш кестелер деп атайды.

Сөздік қосымшасын құру үшін бірнеше тәсіл пайдаланылды. Бірінші литерал әдісі.

Литерал (ағылш. Literal) - тіркелген мәні болып табылатын компьютерлік бағдарламаның бастапқы кодындағы жазба. Кейбір мәліметтер типінің мәндерін бейнелеу немесе объект құрайтын өрнек[3,4].

Литерал әдісін қолдану:

```
>>> d = {}
>>> d
{}
>>> d = {'dict': 1, 'dictionary': 2}
>>> d
{'dict': 1, 'dictionary': 2}
```

Екіншіден dict функциясын қолдану:

```
>>> d = dict(short='dict', long='dictionary')
>>> dPython
{'short': 'dict', 'long': 'dictionary'}
>>> d = dict([(1, 1), (2, 4)])
>>> d
{1: 1, 2: 4}
```

Үшіншіден fromkeys тәсілін қолдану:

```
>>> d = dict.fromkeys(['a', 'b'])
>>> d
{'a': None, 'b': None}
```

```
>>> d = dict.fromkeys(['a', 'b'], 100)
```

```
>>> d
```

```
{'a': 100, 'b': 100}
```

Төртіншіден, тізім генераторларына өте ұқсас сөздік генераторларын қолдану.

```
>>> d = {a: a ** 2 for a in range(7)}
```

```
>>> d
```

```
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36}
```

Енді сөздікке жазбаларды қосып, негізгі мәндерді шығарып көрейік

```
>>> d = {1: 2, 2: 4, 3: 9}
```

```
>>> d[1]
```

```
2
```

```
>>> d[4] = 4 ** 2
```

```
>>> d
```

```
{1: 2, 2: 4, 3: 9, 4: 16}
```

```
>>> d['1']
```

Traceback (most recent call last):

```
File "", line 1, in
```

```
d['1']
```

KeyError: '1'

Мысалдан көріп отырғаныңыздай, жаңа кілт тағайындау сөздікті кеңейтеді, бар кілттің тағайындалуы оны қайта жазады, ал жоқ кілтті алуға тырысу ерекшелік тудырады. Ерекше жағдайды болдырмау үшін арнайы әдіс бар (төменде қараңыз), немесе сіз ерекше жағдайды ала аласыз.

Сөздіктермен тағы не жасауға болады? Ия, басқа объектілермен бірдей: кіріктірілген функциялар, кілт сөздер (мысалы, for және while циклдары), сонымен қатар арнайы сөздік әдістері[1].

### ***Сөздік құруға қажетті әдістер***

`dict.clear()` - сөздікті тазарту.

`dict.copy()` - сөздік көшірмесін қайтарады.

`classmethod dict.fromkeys(seq[, value])` - `seq` кілтті мен `value` мәні бар сөздік құрады (`None` үнсіз келісім).

`dict.get(key[, default])` - сөздікте кілттің мәнін қайтарады, егер кілт болмаса ерекше жағдай орындалмайды, `default` мәні қайтарылады (үнсіз келісім-`None`).

`dict.items()` - жұпты қайтарады (кілт, мән).

`dict.keys()` - сөздікте кілтті қайтарады.

`dict.pop(key[, default])` - кілтті жояды және бастапқы мәнін қайтарады. Егер кілт болмаса, онда `default` мәнін қайтарады (үнсіз келісім бойынша орындайды).

`dict.popitem()` - жұпты жояды және қайтарады (кілт, мән). Егер сөздік бос болса, `KeyError` ерекше жағдайын қалдырады.

`dict.setdefault(key[, default])` - кілттің бастапқы мәні қалады, егер кілт мәні болмаса, `default` мәні үнсіз келісім бойынша орындалады. (`None`-үнсіз келісім).

dict.update([other]) - сөздікті жаңартады, жұп мән қосады (кілт, мән), қолданыстағы кілттер қайта жазылады.

dict.values() - сөздікте мәнді қайтарады[2,3]

Сөздік бойынша жұмыс жасағанда, жиі өздігінен аудару көбінесе түсініксіз түсіндірулерге әкеліп соқтырады. Техникалық сөздіктерде іздеу көп уақытты қажет етеді. Электрондық сөздік қосымшасымен іздеу уақытты қысқартады және аяқталған нәтижеге қол жеткізеді, бірақ әдетте бұл тек тұрақты сөздердің мәнін жеткізбейтін жеке сөздердің аудармасы ғана болады.

Қазіргі кезде технологиялар саласы қарқынды дамып, тез өзгеріске ұшырайтын болғандықтан компьютерлік терминологиялық сөздерді жаңартып отырған жөн.

### Әдебиеттер тізімі

- 1.Kozhayeva S.; Rakhimzhanova M., Ibrayeva K., Muratova G., Dzhumagalieva A. /Formation of humanitarian qualities among students in higher education institutions / S.Kozhayeva, M.Rakhimzhanova, K.Ibrayeva,G.Muratova, A.Dzhumagalieva // Astra Salvensis .-2019.- Issue 13.- p309-326. 18p
- 2.Баймуратова У.С. Электронный инструментарий переводчика. - Оренбург, 2013.-104 с.
- 3.Oxford Advanced Learner's Dictionary. <http://oxforddictionaries.com>
- 4.Загорюлько Ю.А., Боровикова О.И., Загорюлько Г.Б. Построение многоязычных тезаурусов средствами семантической технологии. Институт систем информатик и имени А.П. Ершова СО РАН.- Новосибирск. Материалы конференции OSTIS-2012, 12-16 февраля 2012г. [www.ostis.com](http://www.ostis.com)