

Дәріс №9. Массивтер. Бірөлшемді массив.

Массив – жадта ретімен сақталатын және жалпы деректер типі бар элементтердің жиынтығы. Оның әрбір элементіне оның мәнін оқу немесе өзгерту үшін оған қол жеткізуге мүмкіндік беретін жеке индекс тағайындалады.

Python-дағы массивтердің басты артықшылығы - олардың әртүрлі деректер түрлерін сақтау мүмкіндігі. Бұл оларды әртүрлі мақсаттарда қолдануға мүмкіндік береді: қарапайым деректерді сақтаудан күрделі математикалық операциялар мен алгоритмдерге дейін. Олар сондай-ақ элементтердің кез келген санын сақтай алады, бұл оларды өте икемді және әртүрлі деректер көлемімен жұмыс істеуге ыңғайлы етеді.

Python тіліндегі массивтердің негізгі қолдануларына деректердің үлкен көлемімен жұмыс істеу, матрицаларды жасау және пайдалану, сұрыптау мен іздеуді орындау, әртүрлі алгоритмдер мен деректер құрылымдарын енгізу жатады. Осылайша, олар көптеген бағдарламалық жобалардың ажырамас бөлігі бола отырып, ақпаратты сақтауға, өңдеуге және өңдеуге мүмкіндік береді.

Болашақта тізімдермен жұмысты талдау кезінде біз «массив» сөзін қолданамыз, өйткені көбінесе тізімдер дәл массив ретінде қолданылады (олар бір типтегі деректерді сақтайды).

Массивті құру кезінде жадта бос орын (ұяшықтардың белгілі бір саны) бөлінеді.

1) Массивтерді элементтерді жай тізімдеу арқылы жасауға болады :

```
A = [1, 4, 7, 12, 5]
```

```
print(type(A))
```

Нәтиже

```
<class 'list'>
```

Яғни, массив типті объект болып табылады list (ағылшын тілінен аударылған список).

2) Массивтер кез келген типтегі мәліметтерден – бүтін немесе нақты сандардан, символдық жолдардан құралуы мүмкін.

```
A = [ «alma» , «shie» , «orik» ]
```

Массивтермен жұмыс істеудің кейбір тәсілдері

3) Массивтерді «бүктеуге» болады. $A = [1, 5] + [25, 7]$

4) Бірдей массивтерді қосуды көбейту арқылы ауыстыруға болады. Осылайша бірдей мәндермен толтырылған массив жасау оңай, мысалы: $A = [0] * 10 \# 10$ элементтен тұратын массив жасап, оны нөлдермен толтырды

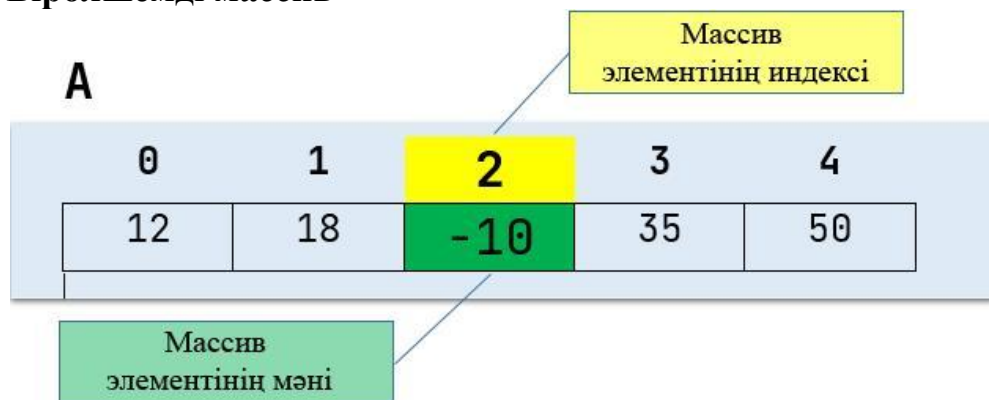
5) Массив әрқашан өзінің өлшемін «біледі». Функция массив өлшемін анықтау үшін `len()` қолданылады. Жиі массив өлшемі бөлек айнымалыда сақталады, сондықтан бағдарламаны басқа массив өлшемімен жұмыс істеу үшін оңай өзгертуге болады.

Мысал

`N = 10` # N айнымалысында біз массивтің өлшемін сақтаймыз
`A = [ 0 ] * N` # N өлшемді массив құру
`print( len ( A ) )` # массив өлшемін көрсетіңіз

Массивтер деректері		
Массивтің түрлеріне қарай:	Массивтің типтері:	Массивтің ерекшелігі:
• бірөлшемді массив; • екіөлшемді массив; • көпөлшемді массив.	• сандық; • мәтіндік; • символдық.	• элементтердің барлығы біртекті; • элементтер бір атаумен аталады; • элементтер жадта қатар орналасады.

Бірөлшемді массив



Массивті таныстыру

`from array import array`
`A = array('i', 5, 7, 11, 25, 29)`

Мысал

Енгізу	Шығару
<code>A = [5, 7, 11, 25, 29]</code>	<code>[5, 7, 11, 25, 29]</code>
<code>print(A)</code>	<code>[5, 11]</code>
<code>print(A[0:4:2])</code>	<code>[11, 25, 29]</code>
<code>print(A[2:])</code>	<code>5 29</code>
<code>print(A[0], A[-1])</code>	<code>[5, 7, 11]</code>
<code>print(A[:3])</code>	<code>[25, 29]</code>
<code>print(A[3:])</code>	

Мысал 1.

Ұзындығы N-ге тең `A[N]` массиві берілген ($N < 100$). Осы массивтегі барлық оң элементтердің қосындысын экранға шығар.

Енгізу	Шығару
3 2 -10 4	6
Түсініктеме коды	<pre> from array import array A=array("i") N=int(input()) S=0 A=[0]*N for i in range(0,N): A[i] = int(input()) for i in range(0,N): if A[i] > 0 : S+=A[i] print(S) </pre>

Мысал 2.

Монеталар

(Уақыты: 1 сек. Жад: 16 МБ Күрделілігі: 8%)

Үстелде N монета бар. Олардың кейбіреулері бас көтереді, ал кейбіреулері бас көтереді. Барлық монеталардың төңкерілуі үшін аударылуы керек монеталардың ең аз санын анықтаңыз.

Енгізу	Шығару
3 1 0 1	1
Түсініктеме коды	<pre> from array import array a = array("i") s1 = 0 s2 = 0 n = int(input()) a = [0] * n for i in range(0,n): a[i] = int(input()) for i in range(0, n): if a[i] > 0: s1 += 1 else: s2 += 1 print(min(s1, s2)) </pre>