

## Блок 1.1. Python тіліне кіріспе

### 1.1.1. Python тілінің анықтамасы

#### Python тілі дегеніміз не?

Python – жоғары деңгейлі, әмбебап бағдарламалау тілі. Ол бағдарламашыларға деректерді өңдеу, автоматтандыру, веб-әзірлеу, ғылыми есептеулер және көптеген басқа да мақсаттар үшін қолданылады. Python тілі қарапайым синтаксисімен, бай стандартты кітапханасымен және модульдер жиынтығымен ерекшеленеді. Бұл тілді 1991 жылы Гвидо ван Россум әзірлеген және бастапқыдан объектіге бағытталған бағдарламалау (Object-Oriented Programming, OOP) негізінде жасалған.

#### Python тілінің негізгі ерекшеліктері

- **Оңай үйрену:** Python-ның синтаксисі қарапайым әрі оқуға түсінікті.
- **Тегін және ашық код:** Python тілін еркін жүктеп, пайдалануға болады.
- **Кросс-платформалық:** Python Windows, macOS және Linux жүйелерінде жұмыс істей береді.
- **Объектіге бағытталған:** Python объектілер мен кластар арқылы бағдарламалау мүмкіндігін береді.
- **Кең қолданыс:** Python деректерді талдау, жасанды интеллект, веб-әзірлеу, графикалық интерфейс жасау сияқты салаларда кеңінен қолданылады.

#### Python-ның даму тарихы

Python тілі 1980 жылдардың соңында әзірлене бастады. Алғашқы ресми нұсқасы 1991 жылдың 20 ақпанында жарық көрді. Бұл күннен бастап Python әлем бойынша тез танылып, көптеген бағдарламашылардың таңдаулы құралына айналды.

#### Python-ның қолдану аясы

Python келесі салаларда кеңінен қолданылады:

- **Веб-әзірлеу:** Django және Flask сияқты фреймворктер арқылы веб-қосымшалар жасау.
- **Ғылыми зерттеулер:** NumPy, SciPy, Matplotlib кітапханалары деректер талдауында және визуализацияда қолданылады.
- **Машиналық оқыту:** TensorFlow және PyTorch сияқты құралдар жасанды интеллект жобаларын әзірлеуде жиі қолданылады.
- **Робототехника:** Python робототехникадағы құрылғыларды бағдарламалау үшін де пайдаланылады.

#### Python тілінің стандартты кітапханасы

Python-ның бай стандартты кітапханасы көптеген дайын шешімдерді қамтиды:

- Файлдармен жұмыс жасау (os, shutil).
- HTTP сұраныстары (requests).
- Деректер визуализациясы (matplotlib).
- Ғылыми есептеулер (numpy, scipy).

## Python бағдарламасын орнату

Python бағдарламасын орнату үшін ресми сайтқа (<https://www.python.org/>) кіріп, қажетті нұсқаны жүктеу керек. Орнату процесі қарапайым және ешқандай арнайы дайындықты қажет етпейді.

## Алғашқы бағдарлама: “Hello, World!”

Python-да бағдарламалау өте қарапайым. Мысалы, келесі кодты іске қосу арқылы алғашқы бағдарламаны жасауға болады:

```
print("Hello, World!")
```

Бұл бағдарлама экранда “Hello, World!” мәтінін көрсетеді.

## Python бағдарламалау тіліне толық сипаттама

### Python-ның негізгі артықшылықтары

Python бағдарламалау тілі өзінің ыңғайлылығы мен әмбебаптығының арқасында заманауи бағдарламашылар арасында үлкен сұранысқа ие. Оның негізгі артықшылықтары:

**Оқуға және жазуға оңай:** Python-ның синтаксисі ағылшын тілінің ережелеріне жақын, бұл бағдарламашылардың кодты тез түсінуіне және жазуына мүмкіндік береді.

**Ауқымды қауымдастық:** Python-ды үйрену барысында немесе белгілі бір мәселені шешу кезінде көп көмекке ие болуға болады, себебі оның пайдаланушыларының қауымдастығы өте үлкен.

**Үздік интеграция:** Python басқа бағдарламалау тілдерімен және құралдармен оңай интеграцияланады (мысалы, C, C++, Java).

**Автоматты басқару:** Жадты басқару Python-да автоматты түрде жүзеге асады, бұл бағдарламашының жұмысын айтарлықтай жеңілдетеді.

### Python-ның стандартты кітапханасы және модульдері

Python бай стандартты кітапханасымен танымал. Бұл кітапхана күрделі функцияларды оңай іске асыруға мүмкіндік береді. Ең танымал модульдер:

- **os:** Файлдық жүйемен жұмыс істеуге арналған.

- **sys:** Бағдарлама аргументтері мен жүйелік функцияларды басқаруға көмектеседі.
- **math:** Математикалық функциялар мен константалар жиынын ұсынады.
- **random:** Кездейсоқ сандарды генерациялау үшін қолданылады.
- **datetime:** Уақыт пен күнді өңдеуге арналған.

## Python-да қолданылатын негізгі функциялар

Функциялар – Python бағдарламалау тілінің негізгі компоненттерінің бірі. Олар кодты қайта-қайта пайдалануға мүмкіндік береді:

- **Басып шығару:** `print()`
- **Кіріс алу:** `input()`
- **Мән типін тексеру:** `type()`
- **Айнымалылар тізімін көру:** `dir()`

## Python тіліндегі негізгі амалдар

### 1. Айнымалыларды енгізу:

```
name = input("Атыңызды енгізіңіз: ")
print("Сәлем, ", name)
```

### 2. Екі айнымалыны алмастыру:

```
a, b = 5, 10
a, b = b, a
print(a, b) # Нәтиже: 10, 5
```

### 3. Қарапайым есептеулер:

```
result = (10 + 5) * 2 / 5
print(result)
```

## Python-ның даму болашағы

Python тек бағдарламашылар арасында ғана емес, ірі компаниялар арасында да кеңінен қолданылады. Google, NASA, Netflix сияқты ұйымдар өз жобаларында Python тілін пайдаланады. Оның болашағы жасанды интеллект, деректер ғылымы және веб-технологиялар сияқты маңызды салалармен тығыз байланысты.

## Python-ды үйренуді қалай бастауға болады?

**1. Қажетті құралдарды орнату:** Python-ның соңғы нұсқасын ресми сайттан жүктеп алу.

**2. Оқу материалдары:** Онлайн курстар, кітаптар, бейне сабақтар арқылы Python-ды зерттеу.

**3. Жаттығулар орындау:** Жаңа үйренген білімді бекіту үшін қарапайым бағдарламалар жазу.

Python – бұл әмбебап және қуатты құрал, оны бағдарламалау әлемінде бастау үшін де, кәсіби деңгейде жұмыс істеу үшін де таңдауға болады. Оның қарапайымдылығы, икемділігі және көп функционалдылығы оны барлық саладағы жобалар үшін ең жақсы тілдердің біріне айналдырады.

### 1.1.2.Сызықты және шартты алгоритмдер

#### **Алгоритмнің анықтамасы**

Алгоритм – бұл белгілі бір есепті шешу үшін орындалатын әрекеттердің жүйелі тізбегі. Алгоритмдер ақпаратты өңдеудің, басқарудың және компьютерлік бағдарламалаудың негізі болып табылады. Олар кез келген мәселені шешудің құрылымды, логикалық және тиімді тәсілін қамтамасыз етеді.

Алгоритмдер тек бағдарламалау саласында ғана емес, күнделікті өмірде де кеңінен қолданылады. Мысалы, тамақ әзірлеу рецепті, құрылғыларды басқару нұсқаулықтары, жолды табу үшін карталарды қолдану – бұлардың бәрі алгоритмдік процестердің қарапайым мысалдары.

#### **Алгоритмнің негізгі қасиеттері:**

**Дәлдік:** Әрбір қадам нақты анықталған болуы керек.

**Түсініктілік:** Алгоритм кез келген орындаушыға түсінікті болуы қажет.

**Аяқталушылық:** Алгоритм белгілі бір қадамнан кейін аяқталуы тиіс.

**Нәтижелілік:** Әрбір алгоритм орындалғаннан кейін нақты нәтиже беруі керек.

**Жалпылық:** Алгоритм тек бір ғана емес, ұқсас типтегі есептерді шешу үшін қолданылады.

#### **Сызықты алгоритмдер**

Сызықты алгоритмдер – бұл қадамдары қатаң реттілікпен орындалатын алгоритмдер. Әрбір әрекет алдыңғы әрекеттен кейін орындалады, ал әрекеттердің орындалу тәртібі

өзгертілмейді. Бұл алгоритмдер құрылымы жағынан қарапайым және түсінікті болып табылады, себебі оларда тармақталу немесе циклдар болмайды. Әрбір әрекет бірінен кейін бірі орындалады, сондықтан оларды орындау барысында тек бір жол ғана қарастырылады.

Сызықты алгоритмдер, әдетте, есептеулерді, деректерді енгізу және шығару сияқты қарапайым операцияларды орындау үшін қолданылады. Мұндай алгоритмдерде қадамдар арасындағы байланыс бір бағытты және логикалық ретке келтірілген. Олардың негізгі ерекшелігі – алгоритмді орындау кезінде ешқандай қосымша шарт немесе қайталау қажет емес.

### **Сызықты алгоритмнің негізгі ерекшеліктері:**

- Барлық қадамдар қатаң реттілікпен орындалады.
- Шарттарды тексеру немесе циклдер жоқ.
- Есептеу, мәлімет енгізу және шығару сияқты қарапайым операцияларда қолданылады.

### **Шартты алгоритмдер**

**Шартты алгоритмдер** – бұл шешім қабылдау процесін қамтитын алгоритмдер.

Мұндай алгоритмдерде белгілі бір шарт тексеріледі, және сол шарттың орындалуына байланысты әрекеттер әртүрлі жолмен орындалуы мүмкін. Егер шарт дұрыс болса (ақиқат), бір әрекет орындалады, ал егер шарт жалған болса – басқа әрекет орындалады немесе әрекет мүлдем орындалмайды.

Шартты алгоритмдер тармақталу құрылымына негізделген және есептерді шешудің икемді тәсілдерін қамтамасыз етеді. Олар тексерілетін шартқа сәйкес, бірнеше мүмкін нұсқалар арасынан ең қолайлы шешімді таңдап, есепті дұрыс орындауға мүмкіндік береді.

### **Шартты алгоритмнің негізгі ерекшеліктері:**

- Бірнеше тармақтан тұруы мүмкін.
- Логикалық шарттар негізінде әрекеттер орындалады.
- Күрделі процестерді модельдеуге мүмкіндік береді.

### **Шартты алгоритмнің құрылымы:**

- Егер шарт дұрыс болса, онда бір әрекет орындалады.
- Егер шарт дұрыс болмаса, басқа әрекет орындалады (немесе орындалмайды).

### **Сызықты және шартты алгоритмдерді салыстыру**

<b>Сипаттамасы</b>	<b>Сызықты алгоритмдер</b>	<b>Шартты алгоритмдер</b>
<b>Құрылымы</b>	Қадамдар реттілікпен орындалады	Шешім қабылдау арқылы бірнеше тармақ бар
<b>Мысалы</b>	Қосынды есептеу, ауданды табу	Санның жұп/тақ екенін анықтау
<b>Қолдану саласы</b>	Қарапайым есептер	Логикалық шарттар қажет болған есептер
<b>Артықшылығы</b>	Қарапайым және жылдам	Күрделі процестерді шешуге қолайлы
<b>Кемшілігі</b>	Тармақталу мен шешім қабылдауға жарамсыз	Күрделі логикалық тексерулерді қажет етеді

Сызықты және шартты алгоритмдер – алгоритмдеу мен бағдарламалаудың негізі. Сызықты алгоритмдер қарапайым, ретті әрекеттерді орындауға тиімді, ал шартты алгоритмдер шешім қабылдауға, тармақталған процестерді модельдеуге мүмкіндік береді. Екі алгоритм түрін дұрыс қолдану кез келген бағдарламалық тапсырманың сәтті шешіміне жетелейді.