

Temel Programlama 1

Öğr. Gör. Furkan DURMUŞ

Samsun Üniversitesi
Teknik Bilimler
Meslek Yüksekokulu



Bu Haftanın Ders Kazanımları

Bu haftayı bitirdiğinizde,

{ 1 } Algoritma Nedir?

{ 2 } Sahte Kod

{ 3 } Akış Diyagramı

{ 4 } Neden Python

{ 5 } Python Kurulumu

{ 6 } IDE Kurulumu

öğrenmiş olacaksınız.

ALGORITMA

```
0 response = requests.get(url) # download from the website
1
2 # checking response.status_code (if you get 502, try rerunning the code)
3 if response.status_code != 200:
4     print('Status: {response.status_code} - Try rerunning the code\n')
5     print('Status: {response.status_code}\n')
6
7 # using BeautifulSoup to parse the response object
8 soup = BeautifulSoup(response.content, "html.parser")
9
10 # finding Post images in the soup
11 find all("img", attrs={"alt": "Post image"})
```

{01} Algoritma Nedir ?

Algoritma sözcüğü, Özbekistan 'ın Harezmi, bugünkü Türkmenistan 'ın Khiva kentinde doğmuş olan Ebu Abdullah Muhammed İbn Musa el Harezmi adındaki İran'lı müslüman alimden gelmektedir.

Bu alim 9. yüzyılda cebir alanındaki algoritmik çalışmalarını kitaba dökerek matematiğe çok büyük bir katkı sağlamıştır.

"Hisab el-cebir" ve "el-mukabala« kitabı dünyanın ilk cebir kitabı ve aynı zamanda ilk algoritma koleksiyonunu oluşturur.



{01} Algoritma Nedir ?



Latince çevirisi Avrupa'da çok ilgi görür. Alimin ismini telaffuz edemeyen Avrupalılar "**algorizm**" sözcüğünü "Arap sayıları kullanarak aritmetik problemler çözme kuralları" manasında kullanırlar.

Bu sözcük daha sonra "**Algoritma**"ya dönüşür ve genel kapsamda kullanılır.

{01} Algoritma Nedir ?



Algoritma, verilen herhangi bir sorunun çözümüne ulaşmak için uygulanması gerekli adımların **hiçbir yoruma yer vermeksizin** açık, düzenli ve sıralı bir şekilde söz ve yazı ile ifadesidir.

{01} Algoritmanın Temel Özellikleri



{01} Başlangıç ve bitiş noktası belirli ve açık olmalıdır.

- Algoritma, açık bir başlangıç noktası ile başlamalı ve sonunda kesin olarak sona erdirilmelidir.

{01} Algoritmanın Temel Özellikleri



{02} Her türlü durumu kapsamalıdır.

- Algoritma, problemin çözümüne ilişkin tüm olası durumları göz önünde bulundurmalı ve eksiksiz çalışmalıdır.

{01} Algoritmanın Temel Özellikleri



{03} Genellik (Aynı Tür Problemler İçin Geçerlilik)

- Algoritma, aynı problem türü için farklı veriler üzerinde çalışabilir olmalı.

{01} Algoritmanın Temel Özellikleri



{04} Sonluluk

- Algoritma, adımları sonlu sayıda olup, bir noktada tamamlanmalıdır.

{01} Algoritmanın Temel Özellikleri



{05} Belirlilik

- Her adım net ve anlaşılır şekilde tanımlanmalıdır.

{01} Algoritmanın Temel Özellikleri



{06} Girdi ve Çıktı

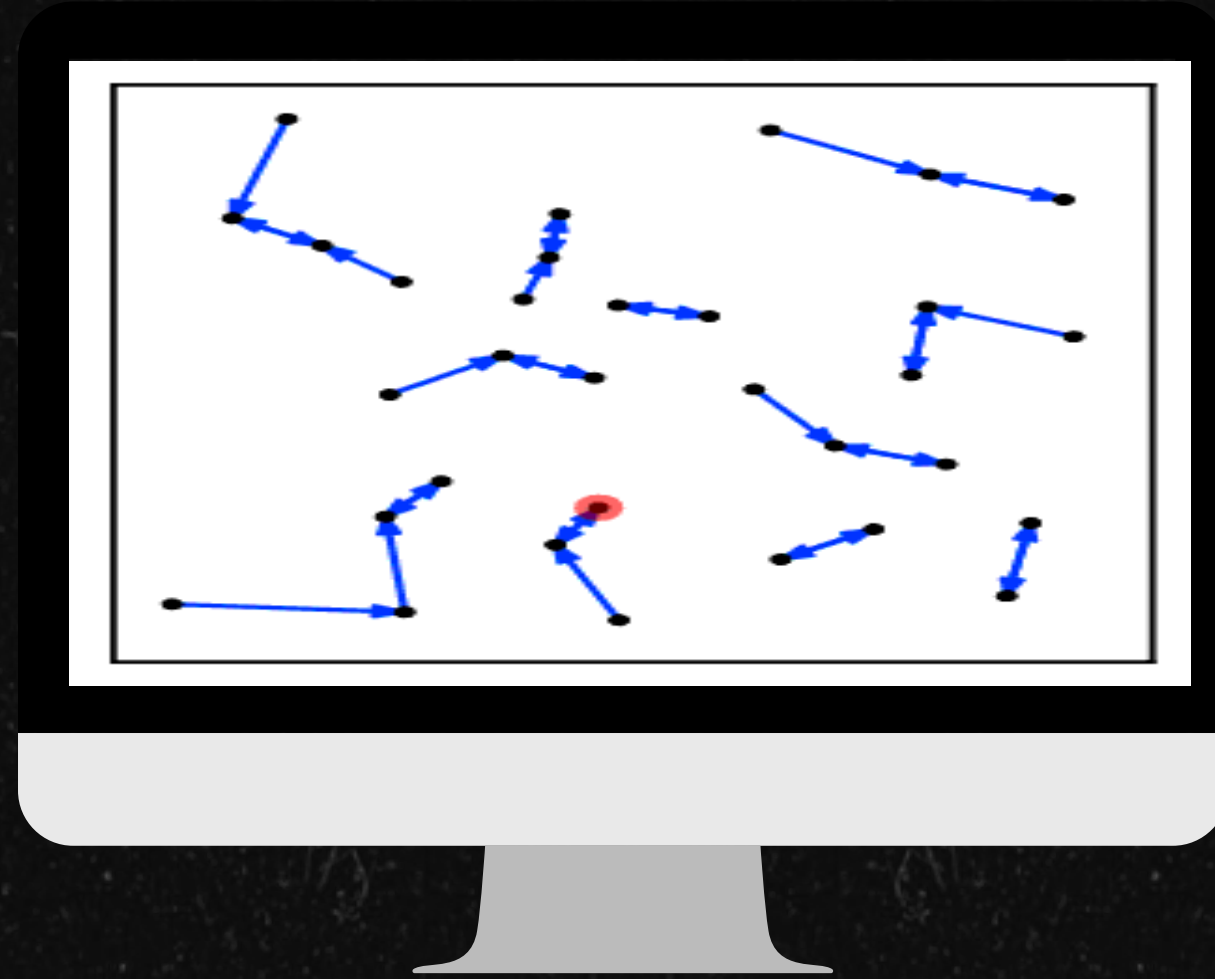
- Algoritma, sıfır veya daha fazla girdi alabilir ve en az bir çıktı üretmelidir.

{01} Algoritmanın Temel Özellikleri



{07} Uygulanabilirlik

- Adımlar uygulanabilir ve mantıksal olarak yapılabilir olmalıdır.



Algoritma Tasarımı

Öğretmeni size 100 vermek için programlayacak algoritmayı yazınız.

- 1 - Başla
- 2 - Web tarayıcıyı aç
- 3 - Okulun web sayfasını aç
- 4 - Akademisyen girişine tıkla
- 5 - Kullanıcı adı ve şifreni gir
- 6 - Giriş butonuna tıkla
- 7 - Dersler sekmesine tıkla
- 8 - "Temel Programlama 1" dersine tıkla
- 9 - "Not Gir" sekmesine tıkla
- 10 - Öğrenci ismi kısmına "Furkan" yaz
- 11 - Öğrenci no kısmına "103" yaz
- 12 - Not kısmına "100" yaz
- 13 - " Kaydet " butonuna tıkla
- 14 - Kaydettiğinden emin ol
- 15 - Bitir

Algoritmanın Analizi

- 1 - Başla
- 2 - Web tarayıcıyı aç
- 3 - Okulun web sayfasını aç
- 4 - Akademisyen girişine tıkla
- 5 - Kullanıcı adı ve şifreni gir
- 6 - Giriş butonuna tıkla
- 7 - Dersler sekmesine tıkla
- 8 - "Temel Programlama 1" dersine tıkla
- 9 - "Not Gir" sekmesine tıkla
- 10 - Öğrenci ismi kısmına "Furkan" yaz
- 11 - Öğrenci no kısmına "103" yaz
- 12 - Not kısmına "100" yaz
- 13 - " Kaydet " butonuna tıkla
- 14 - Kaydettiğinden emin ol
- 15 - Bitir

{01}

Başlangıç ve bitiş noktası belirli ve açık olmalıdır.

- Algoritma, açık bir başlangıç noktası ile başlamalı ve sonunda kesin olarak sona erdirilmelidir.

Algoritmanın Analizi

- 1 - Başla
- 2 - Web tarayıcıyı aç
- 3 - Okulun web sayfasını aç
- 4 - Akademisyen girişine tıkla
- 5 - Kullanıcı adı ve şifreni gir
- 6 - Giriş butonuna tıkla
- 7 - Dersler sekmesine tıkla
- 8 - "Temel Programlama 1" dersine tıkla
- 9 - "Not Gir" sekmesine tıkla
- 10 - Öğrenci ismi kısmına "Furkan" yaz
- 11 - Öğrenci no kısmına "103" yaz
- 12 - Not kısmına "100" yaz
- 13 - " Kaydet " butonuna tıkla
- 14 - Kaydettiğinden emin ol
- 15 - Bitir

{02}

Her türlü durumu kapsamalıdır.

- Algoritma, problemin çözümüne ilişkin tüm olası durumları göz önünde bulundurmalı ve eksiksiz çalışmalıdır.

Algoritmanın Analizi

- 1 - Başla
- 2 - Web tarayıcıyı aç
- 3 - Okulun web sayfasını aç
- 4 - Akademisyen girişine tıkla
- 5 - Kullanıcı adı ve şifreni gir
- 6 - Giriş butonuna tıkla
- 7 - Dersler sekmesine tıkla
- 8 - "Temel Programlama 1" dersine tıkla
- 9 - "Not Gir" sekmesine tıkla
- 10 - Öğrenci ismi kısmına "Furkan" yaz
- 11 - Öğrenci no kısmına "103" yaz
- 12 - Not kısmına "100" yaz
- 13 - " Kaydet " butonuna tıkla
- 14 - Kaydettiğinden emin ol
- 15 - Bitir

{03}

Genellik (Aynı Tür Problemler İçin Geçerlilik)

- Algoritma, aynı problem türü için farklı veriler üzerinde çalışabilir olmalı.

Algoritmanın Analizi

- 1 - Başla
- 2 - Web tarayıcıyı aç
- 3 - Okulun web sayfasını aç
- 4 - Akademisyen girişine tıkla
- 5 - Kullanıcı adı ve şifreni gir
- 6 - Giriş butonuna tıkla
- 7 - Dersler sekmesine tıkla
- 8 - "Temel Programlama 1" dersine tıkla
- 9 - "Not Gir" sekmesine tıkla
- 10 - Öğrenci ismi kısmına "Furkan" yaz
- 11 - Öğrenci no kısmına "103" yaz
- 12 - Not kısmına "100" yaz
- 13 - " Kaydet " butonuna tıkla
- 14 - Kaydettiğinden emin ol
- 15 - Bitir

{04}

Sonluluk

- Algoritma, adımları sonlu sayıda olup, bir noktada tamamlanmalıdır.

Algoritmanın Analizi

- 1 - Başla
- 2 - Web tarayıcıyı aç
- 3 - Okulun web sayfasını aç
- 4 - Akademisyen girişine tıkla
- 5 - Kullanıcı adı ve şifreni gir
- 6 - Giriş butonuna tıkla
- 7 - Dersler sekmesine tıkla
- 8 - "Temel Programlama 1" dersine tıkla
- 9 - "Not Gir" sekmesine tıkla
- 10 - Öğrenci ismi kısmına "Furkan" yaz
- 11 - Öğrenci no kısmına "103" yaz
- 12 - Not kısmına "100" yaz
- 13 - " Kaydet " butonuna tıkla
- 14 - Kaydettiğinden emin ol
- 15 - Bitir

{05}

Belirlilik

- Her adım net ve anlaşılır şekilde tanımlanmalıdır.

Algoritmanın Analizi

- 1 - Başla
- 2 - Web tarayıcıyı aç
- 3 - Okulun web sayfasını aç
- 4 - Akademisyen girişine tıkla
- 5 - Kullanıcı adı ve şifreni gir
- 6 - Giriş butonuna tıkla
- 7 - Dersler sekmesine tıkla
- 8 - "Temel Programlama 1" dersine tıkla
- 9 - "Not Gir" sekmesine tıkla
- 10 - Öğrenci ismi kısmına "Furkan" yaz
- 11 - Öğrenci no kısmına "103" yaz
- 12 - Not kısmına "100" yaz
- 13 - " Kaydet " butonuna tıkla
- 14 - Kaydettiğinden emin ol
- 15 - Bitir

{06}

Girdi ve Çıktı

- Algoritma, sıfır veya daha fazla girdi alabilir ve en az bir çıktı üretmelidir.

Algoritmanın Analizi

- 1 - Başla
- 2 - Web tarayıcıyı aç
- 3 - Okulun web sayfasını aç
- 4 - Akademisyen girişine tıkla
- 5 - Kullanıcı adı ve şifreni gir
- 6 - Giriş butonuna tıkla
- 7 - Dersler sekmesine tıkla
- 8 - "Temel Programlama 1" dersine tıkla
- 9 - "Not Gir" sekmesine tıkla
- 10 - Öğrenci ismi kısmına "Furkan" yaz
- 11 - Öğrenci no kısmına "103" yaz
- 12 - Not kısmına "100" yaz
- 13 - " Kaydet " butonuna tıkla
- 14 - Kaydettiğinden emin ol
- 15 - Bitir

{07}

Uygulanabilirlik

- Adımlar uygulanabilir ve mantıksal olarak yapılabilir olmalıdır.

Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

Başla

Alarm çalınca uyan



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

Başla
Alarm çalınca uyan
Yataktan kalk
Yüzünü yıka, dişlerini fırçala



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git  
Okul kapısından içeri gir
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git  
Okul kapısından içeri gir  
Sınıfını bul
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git  
Okul kapısından içeri gir  
Sınıfını bul  
Sınıfa gir
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git  
Okul kapısından içeri gir  
Sınıfını bul  
Sınıfa gir  
Sırana otur
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git  
Okul kapısından içeri gir  
Sınıfını bul  
Sınıfa gir  
Sırana otur  
Derste öğretmeni dinle
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git  
Okul kapısından içeri gir  
Sınıfını bul  
Sınıfa gir  
Sırana otur  
Derste öğretmeni dinle  
Not al
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git  
Okul kapısından içeri gir  
Sınıfını bul  
Sınıfa gir  
Sırana otur  
Derste öğretmeni dinle  
Not al  
Derste verilen etkinlikleri yap
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git  
Okul kapısından içeri gir  
Sınıfını bul  
Sınıfa gir  
Sırana otur  
Derste öğretmeni dinle  
Not al  
Derste verilen etkinlikleri yap  
Ders ile ilgili sorun varsa öğretmene sor
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git  
Okul kapısından içeri gir  
Sınıfını bul  
Sınıfa gir  
Sırana otur  
Derste öğretmeni dinle  
Not al  
Derste verilen etkinlikleri yap  
Ders ile ilgili sorun varsa öğretmene sor  
Ayağa Kalk
```



Algoritma Tasarımı

Birlikte adım adım sabah kalktığımızdan itibaren ders sonuna kadar geçen sürecin algoritmasını yazalım;

```
Başla  
Alarm çalınca uyan  
Yataktan kalk  
Yüzünü yıka, dişlerini fırçala  
Üzerini giyin  
Kahvaltı yap  
Çantayı al  
Evden çık  
Okula git  
Okul kapısından içeri gir  
Sınıfını bul  
Sınıfa gir  
Sırana otur  
Derste öğretmeni dinle  
Not al  
Derste verilen etkinlikleri yap  
Ders ile ilgili sorun varsa öğretmene sor  
Ayağa Kalk  
Ders arasına çık -> Afiyet Olsun ☺
```

Bitiş



Sözde Kod

(Sahte kod-Pseudo code)

```
0 response = requests.get(url) # load from the website
1
2 # checking response.status_code (if you get 502, try rerunning the code)
3 if response.status_code != 200:
4     status: {response.status_code} - Try rerunning the code\n"
5
6 # using BeautifulSoup to parse the response object
7 soup = BeautifulSoup(response.content, "html.parser")
8
9 # finding Post images in the soup
10 find all("img", attrs={"alt": "Post image"})
```

{02} Sahte Kod ve Algoritma

Programlama dili seçseniz bile, hemen kod yazımına geçemezsiniz. “Pseudocode– sahte kod” denilen kodlama ile programın işlemleri yazılır. Sahte kod kurduğunuz mantıktaki eksik yada hatalı noktaları bulmaya yarar.



{02} Sahte Kod ve Algoritma

Örneğin; Eğer bir şehre füze göndermek isterseniz adımlarınız şu şekilde olabilir:

1. Hedefin koordinatlarını al
2. Füzenin şu anki koordinatını al
3. Hedefi vurması için gerekeni hesapla
4. Füzeyi ateşle

1.adımın daha da gelişmiş hâli şöyle olabilir:

- 1.a. Füze teknisyeninden hedef koordinatları edin
- 1.b. Hedef koordinatların geçerli olduğundan emin ol
- 1.c. Hedef koordinatları hafızaya kaydet

1.a. maddesini daha alt basamaklara bölebiliriz.

- 1.a.i. Koordinattan emin ol
- 1.a.ii. Hedef füzenin atış uzaklığı içinde mi, kontrol et
- 1.a.iii. Hedef dost bölge mi, kontrol et



{02} Sahte Kod ve Algoritma

Bu “üstten aşağı– top down” tasarım ile kaba bir taslak gittikçe ayrıntılı şekilde parçalanır. Genelden özele doğru işlemleri belirleriz. Bu zaman kaybı gibi görülebilir.

Plansız olarak programınızı yazmaya başlayabilirsiniz. Ama bu arabaya atlayıp Ankara’dan Elmadağ’a gitmek için kuzeye doğru gitmeye benzer.

Sahte kodlar programınızın ana hatlarını belirtir. Böylece bilgisayarın başaracağı iş için gerekli tüm veriyi toplamış olursunuz.



«Metinsel Algoritma»

Adım 1- Başla
Adım 2- Sayıyı oku
Adım 3- Sayının karesini hesapla
Adım 4- Sonucu ekrana yaz
Adım 5- Bitir

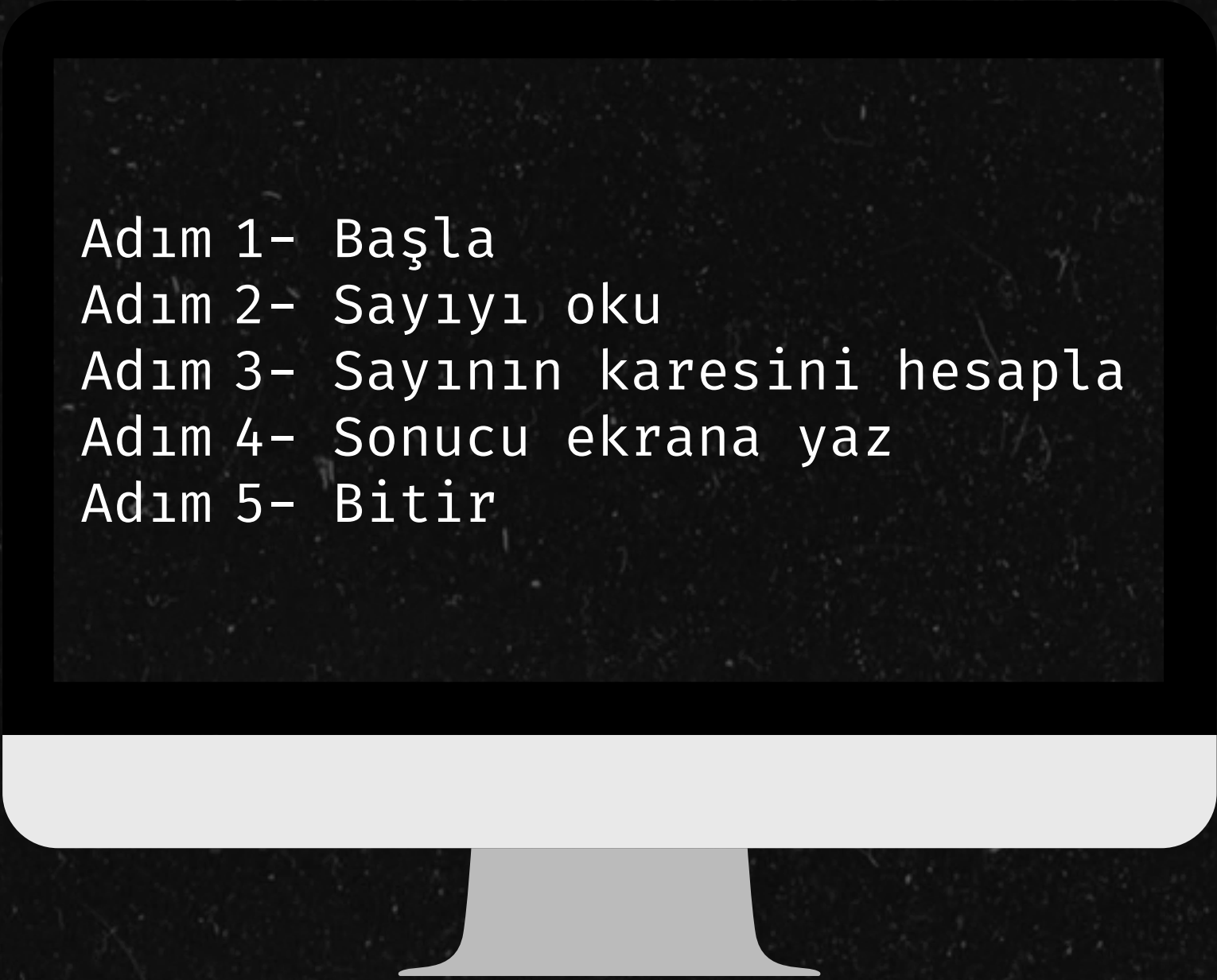
«Sözde Kod»

```
var scrollTop = 0;
element.clientHeight + 0.02
window.scroll(0, scrollTop);
```

Değişken: Her seferinde farklı değerler alabilen ifadelerdir.

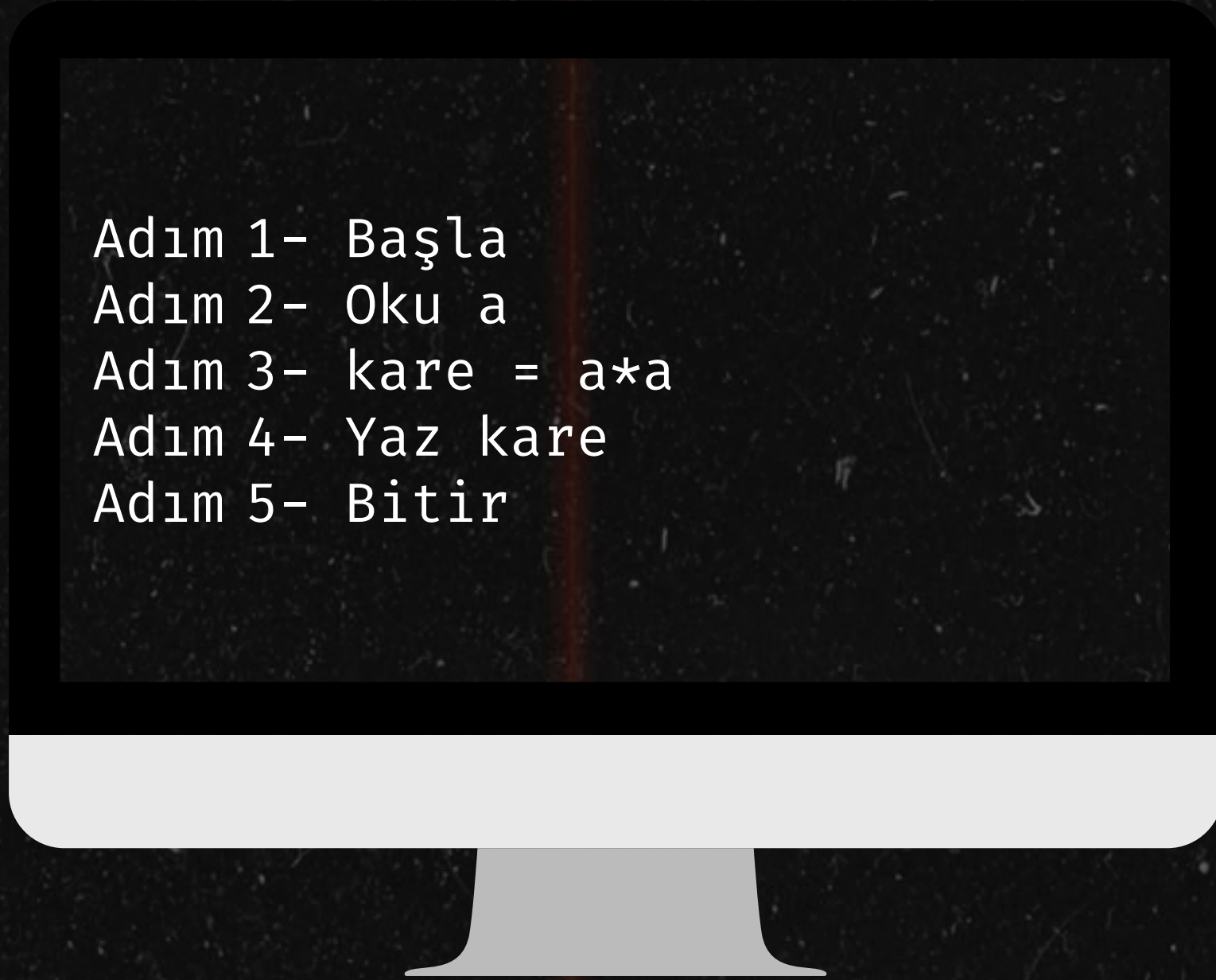
Sabit: Değeri değişmeyen ifadelerdir.

«Metinsel Algoritma»



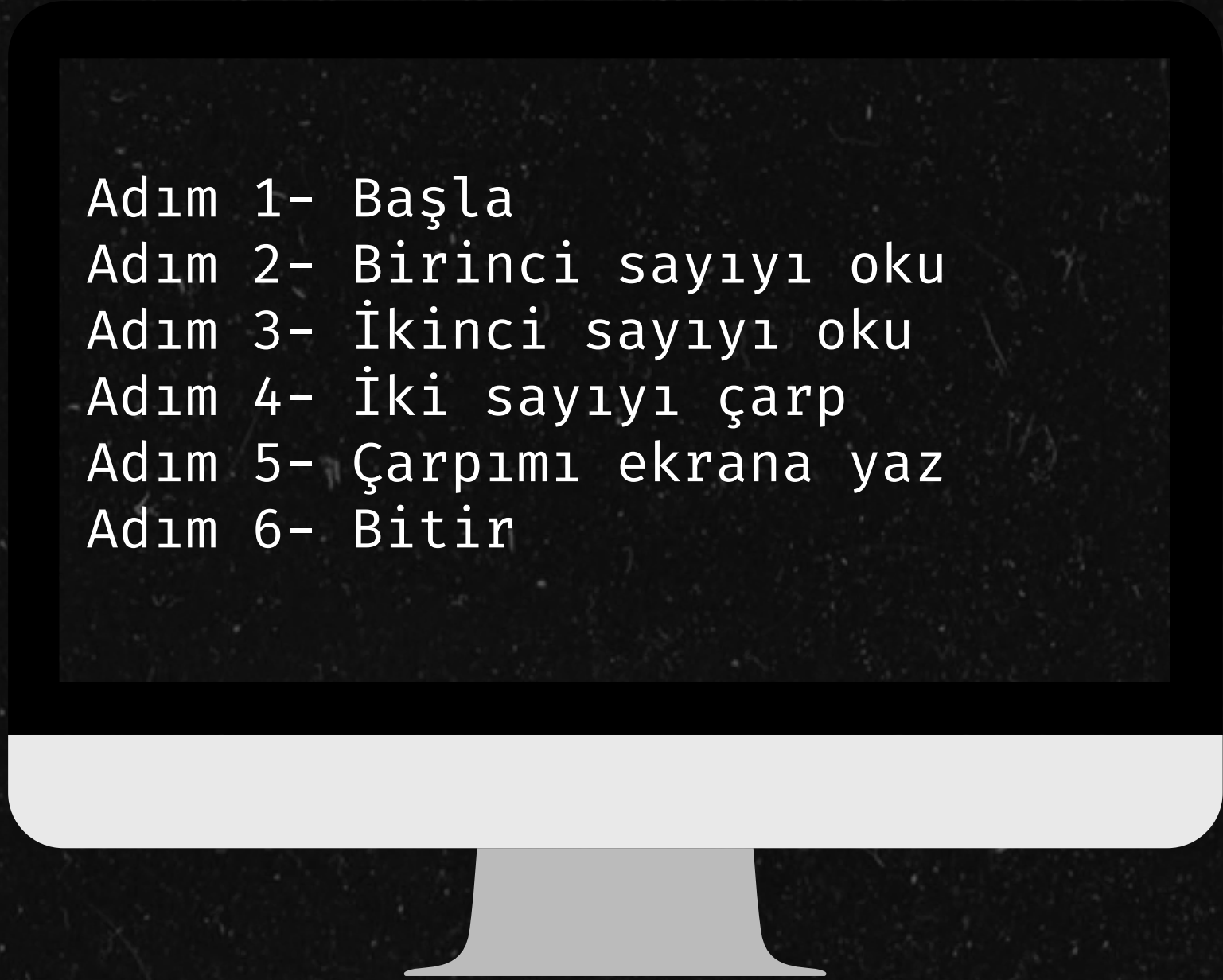
```
Adım 1- Başla  
Adım 2- Sayıyı oku  
Adım 3- Sayının karesini hesapla  
Adım 4- Sonucu ekrana yaz  
Adım 5- Bitir
```

«Sözde Kod»



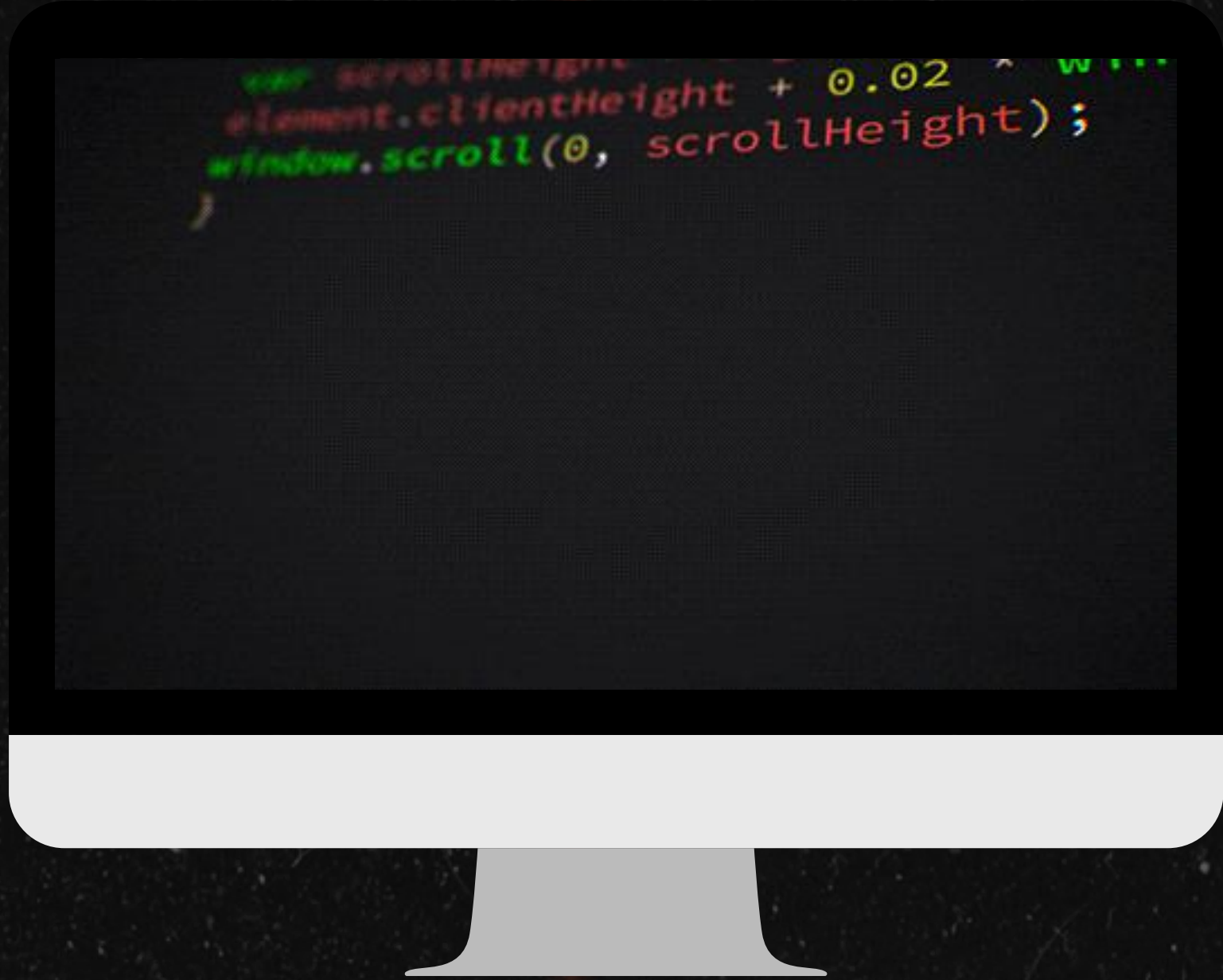
```
Adım 1- Başla  
Adım 2- Oku a  
Adım 3- kare = a*a  
Adım 4- Yaz kare  
Adım 5- Bitir
```

«Metinsel Algoritma»



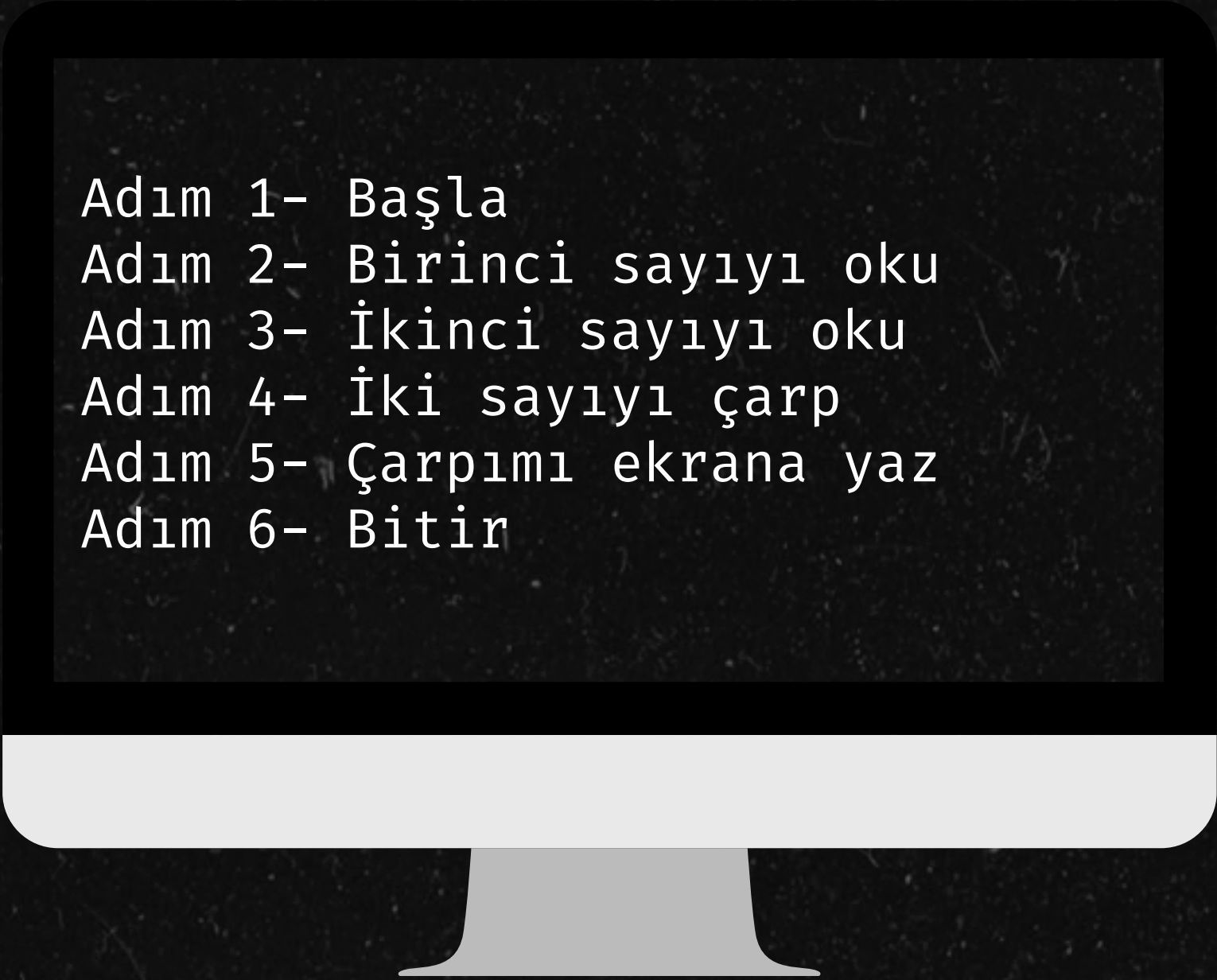
Adım 1- Başla
Adım 2- Birinci sayıyı oku
Adım 3- İkinci sayıyı oku
Adım 4- İki sayıyı çarp
Adım 5- Çarpımı ekrana yaz
Adım 6- Bitir

«Sözde Kod»



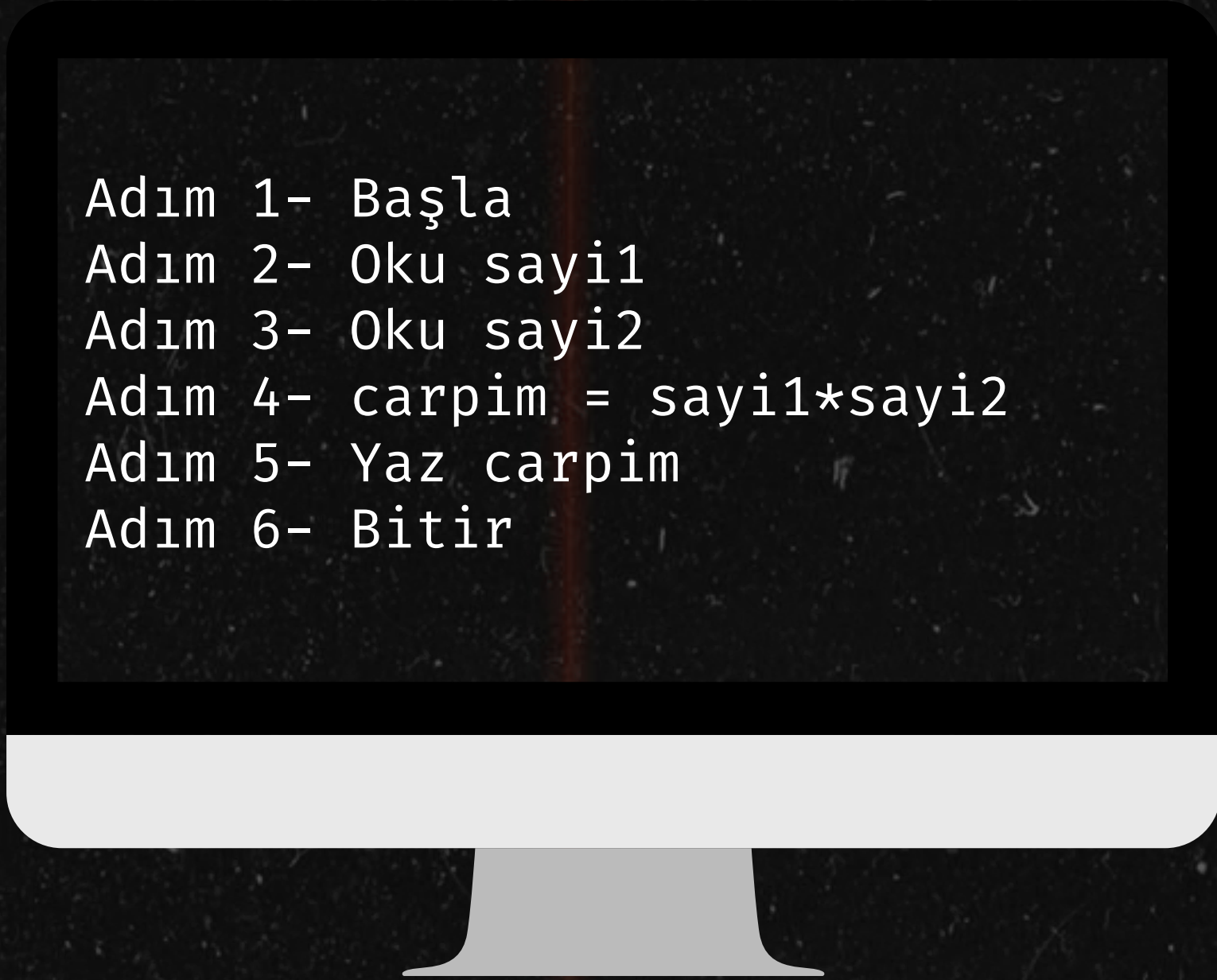
```
var scrollHeight = ...  
element.clientHeight + 0.02  
window.scroll(0, scrollHeight);  
}
```

«Metinsel Algoritma»



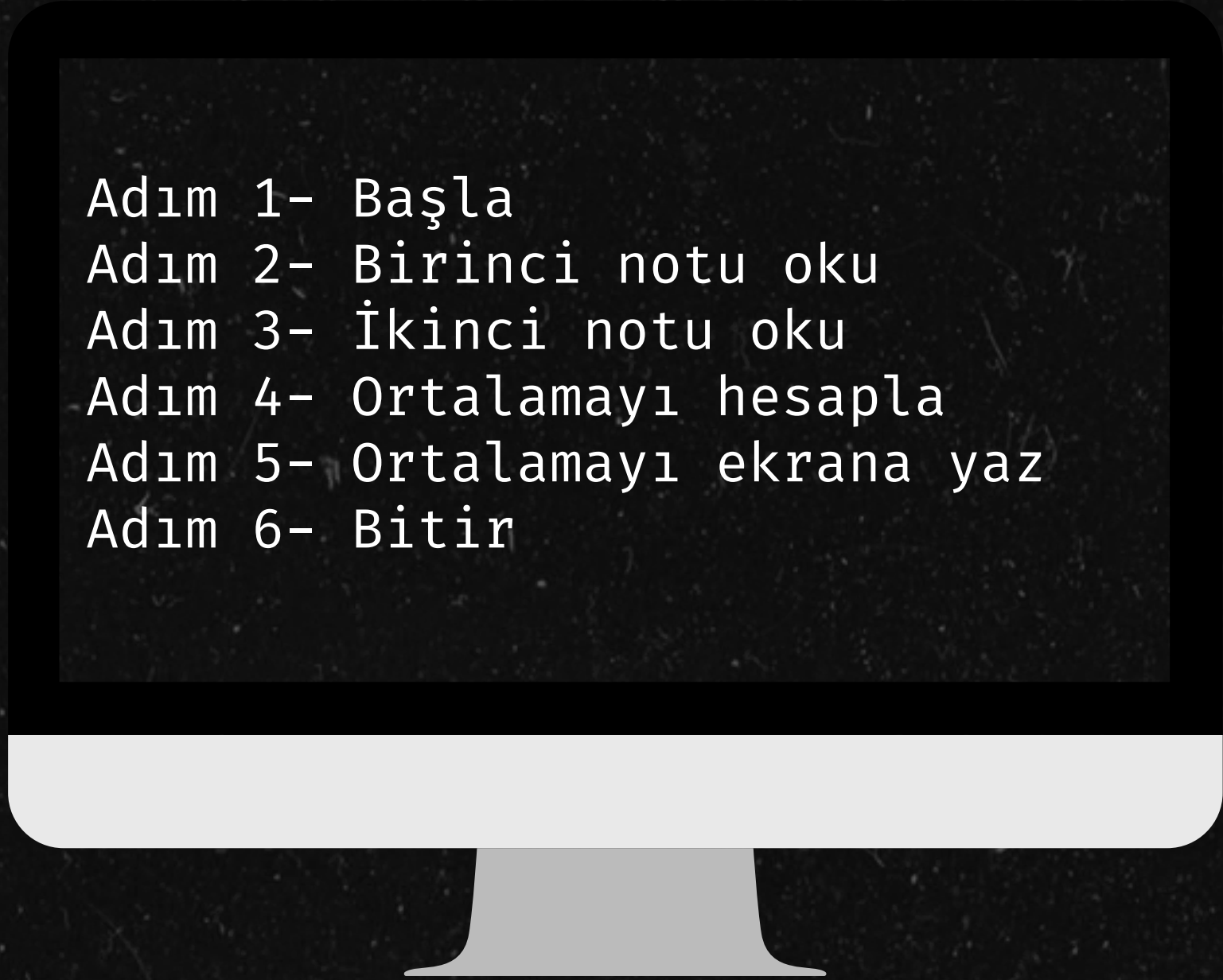
```
Adım 1- Başla  
Adım 2- Birinci sayıyı oku  
Adım 3- İkinci sayıyı oku  
Adım 4- İki sayıyı çarp  
Adım 5- Çarpımı ekrana yaz  
Adım 6- Bitir
```

«Sözde Kod»



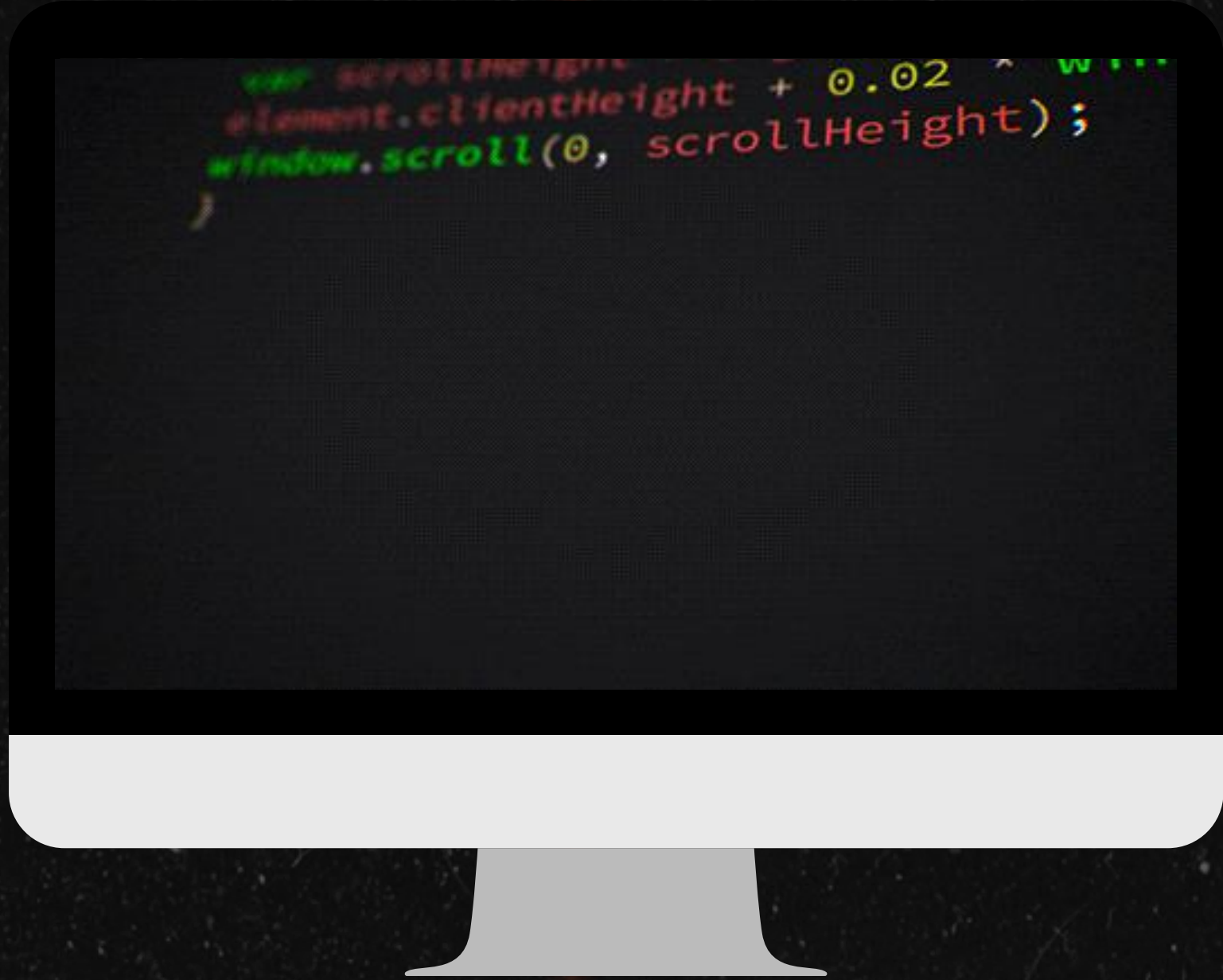
```
Adım 1- Başla  
Adım 2- Oku sayi1  
Adım 3- Oku sayi2  
Adım 4- carpim = sayi1*sayi2  
Adım 5- Yaz carpim  
Adım 6- Bitir
```

«Metinsel Algoritma»



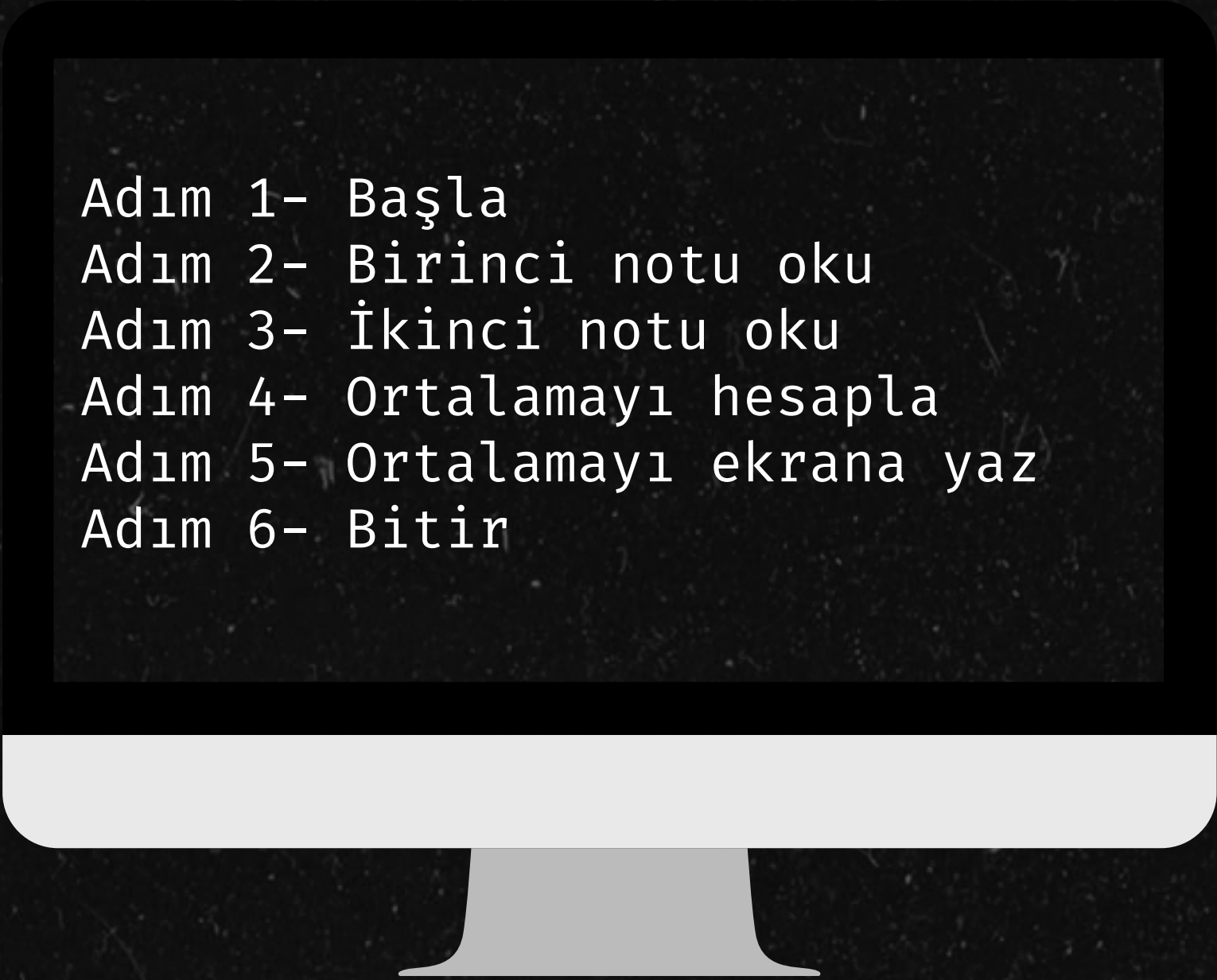
Adım 1- Başla
Adım 2- Birinci notu oku
Adım 3- İkinci notu oku
Adım 4- Ortalamayı hesapla
Adım 5- Ortalamayı ekrana yaz
Adım 6- Bitir

«Sözde Kod»



```
var scrollHeight = ...  
element.clientHeight + 0.02  
window.scroll(0, scrollHeight);  
}
```

«Metinsel Algoritma»



```
Adım 1- Başla  
Adım 2- Birinci notu oku  
Adım 3- İkinci notu oku  
Adım 4- Ortalamayı hesapla  
Adım 5- Ortalamayı ekrana yaz  
Adım 6- Bitir
```

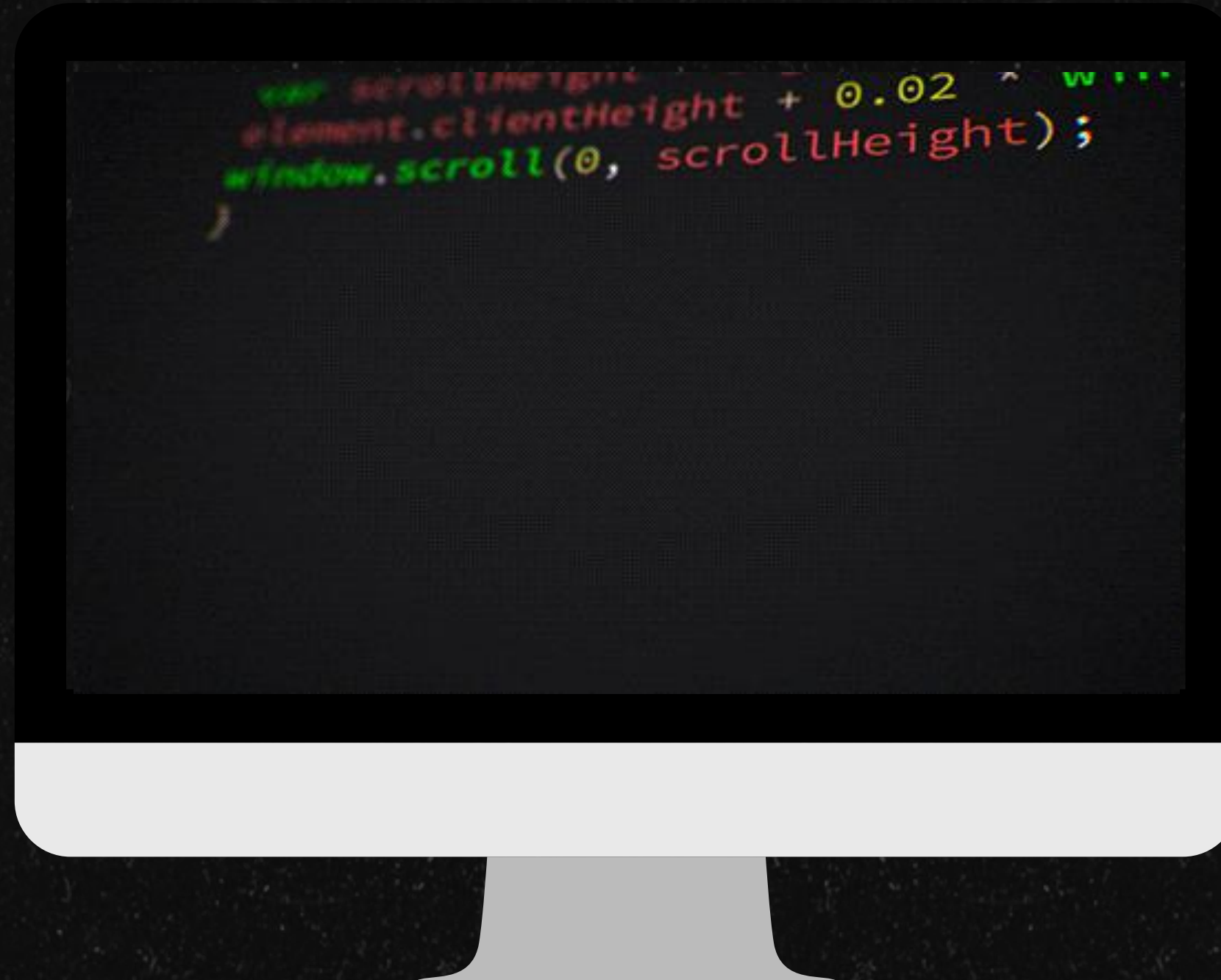
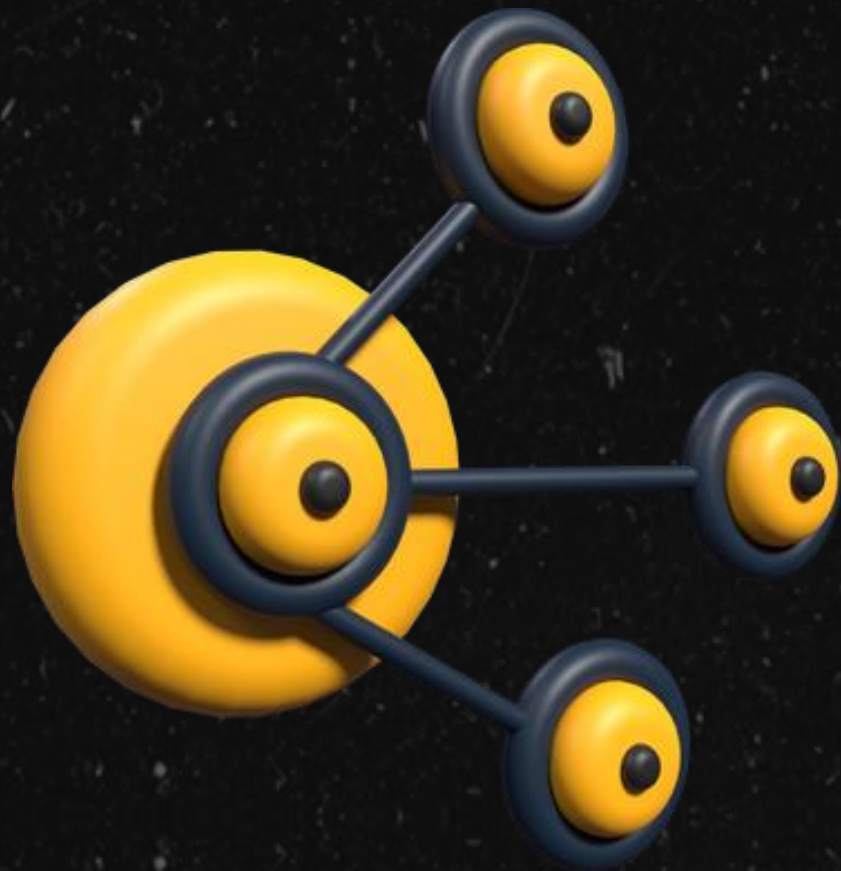
«Sözde Kod»



```
Adım 1- Başla  
Adım 2- Oku not1  
Adım 3- Oku not2  
Adım 4- ortalama=(not1+not2)/2  
Adım 5- Yaz ortalama  
Adım 6- Bitir
```

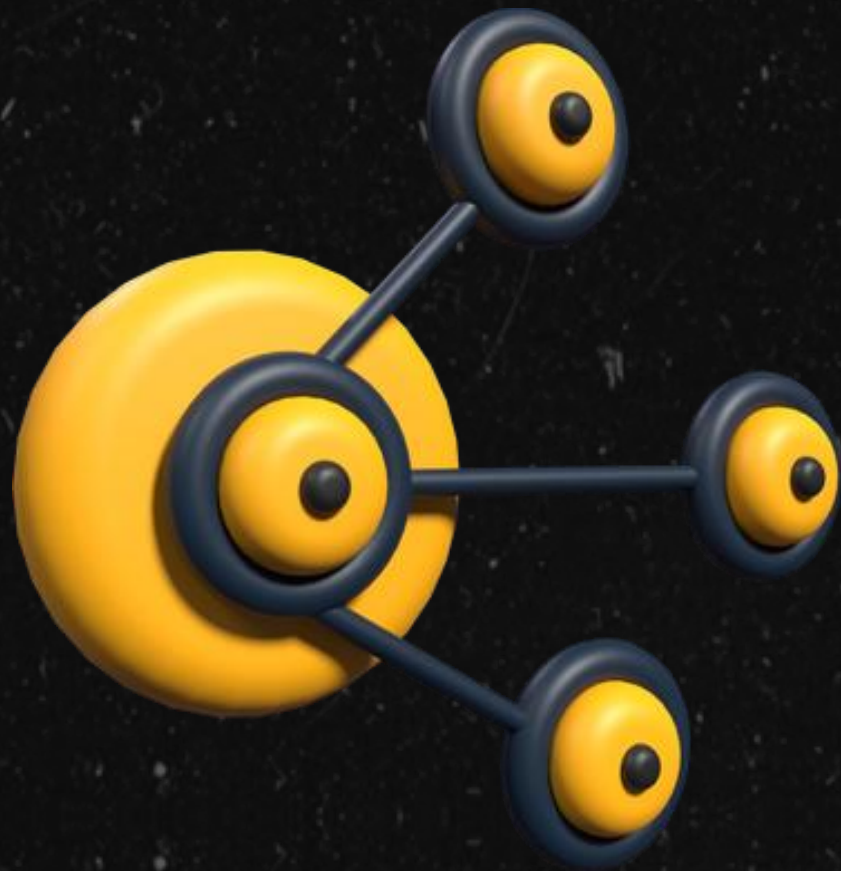
{02} Sahte Kod ve Algoritma

Klavyeden girilen iki sayının büyük olanından küçük olanını çıkaran algoritmayı yazınız.



{02} Sahte Kod ve Algoritma

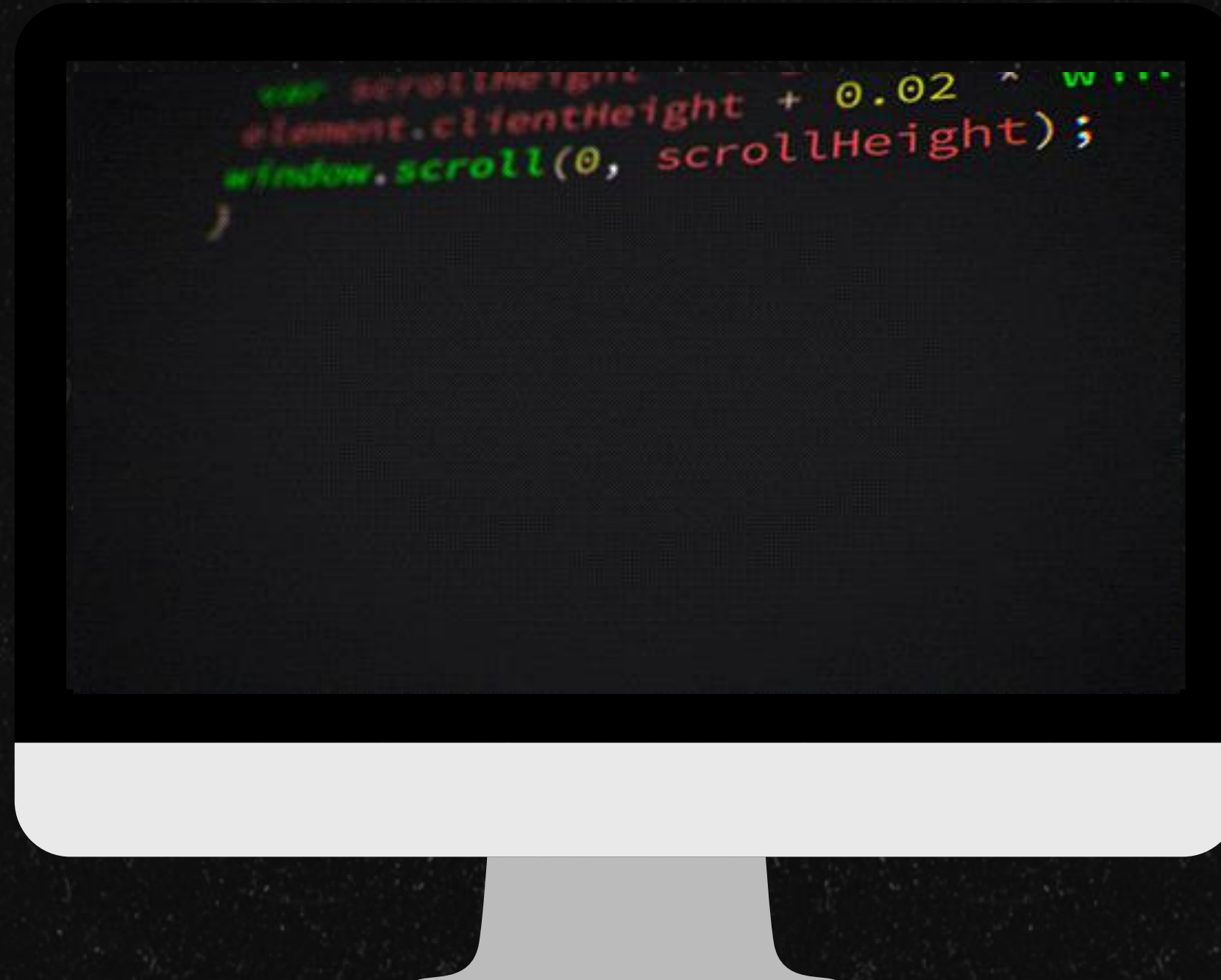
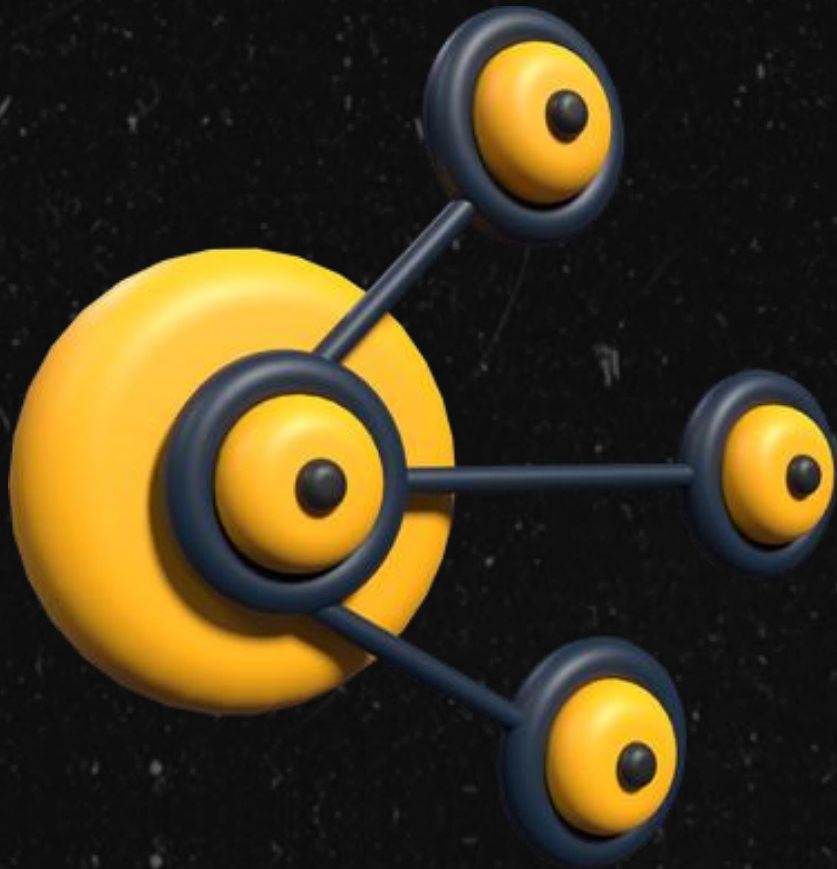
Klavyeden girilen iki sayının büyük olanından küçük olanını çıkaran algoritmayı yazınız.



```
Adım 1- Başla  
Adım 2- Oku sayi1  
Adım 3- Oku sayi2  
Adım 4- Eğer sayi1>sayi2 ise  
           sonuc=sayi1-sayi2  
Adım 5- Değilse sonuc=sayi2-sayi1  
Adım 6- Yaz sonuc  
Adım 7- Bitir
```

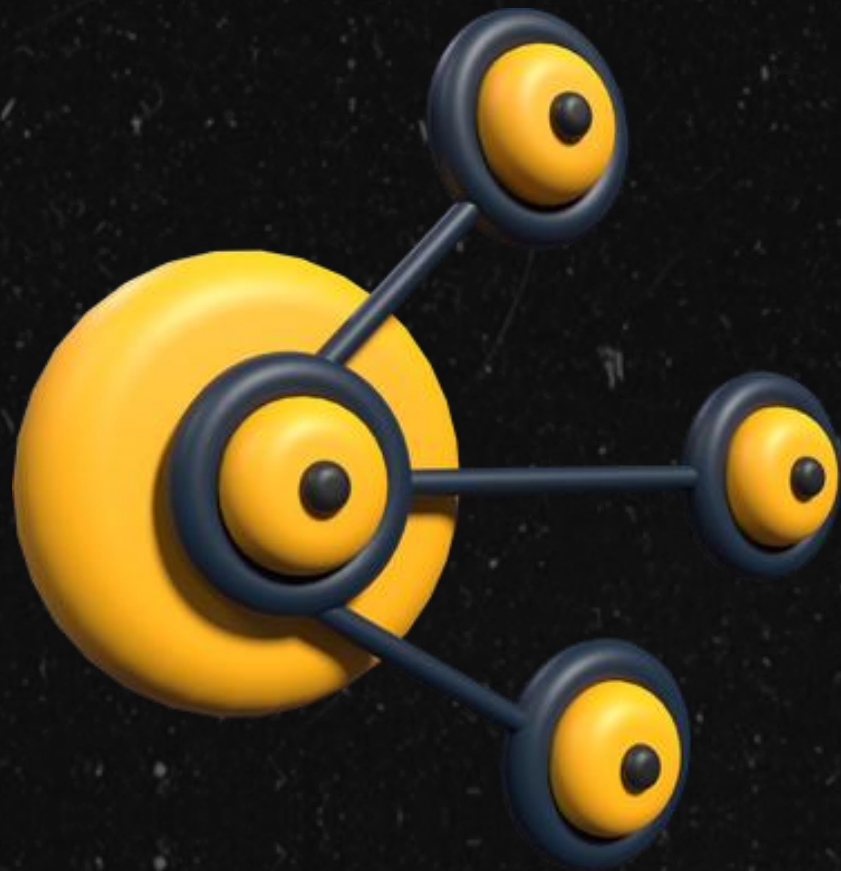
{02} Sahte Kod ve Algoritma

Suyun sıcaklığına göre maddenin katı, sıvı veya gaz olma durumunu gösteren algoritmayı hazır layınız.



{02} Sahte Kod ve Algoritma

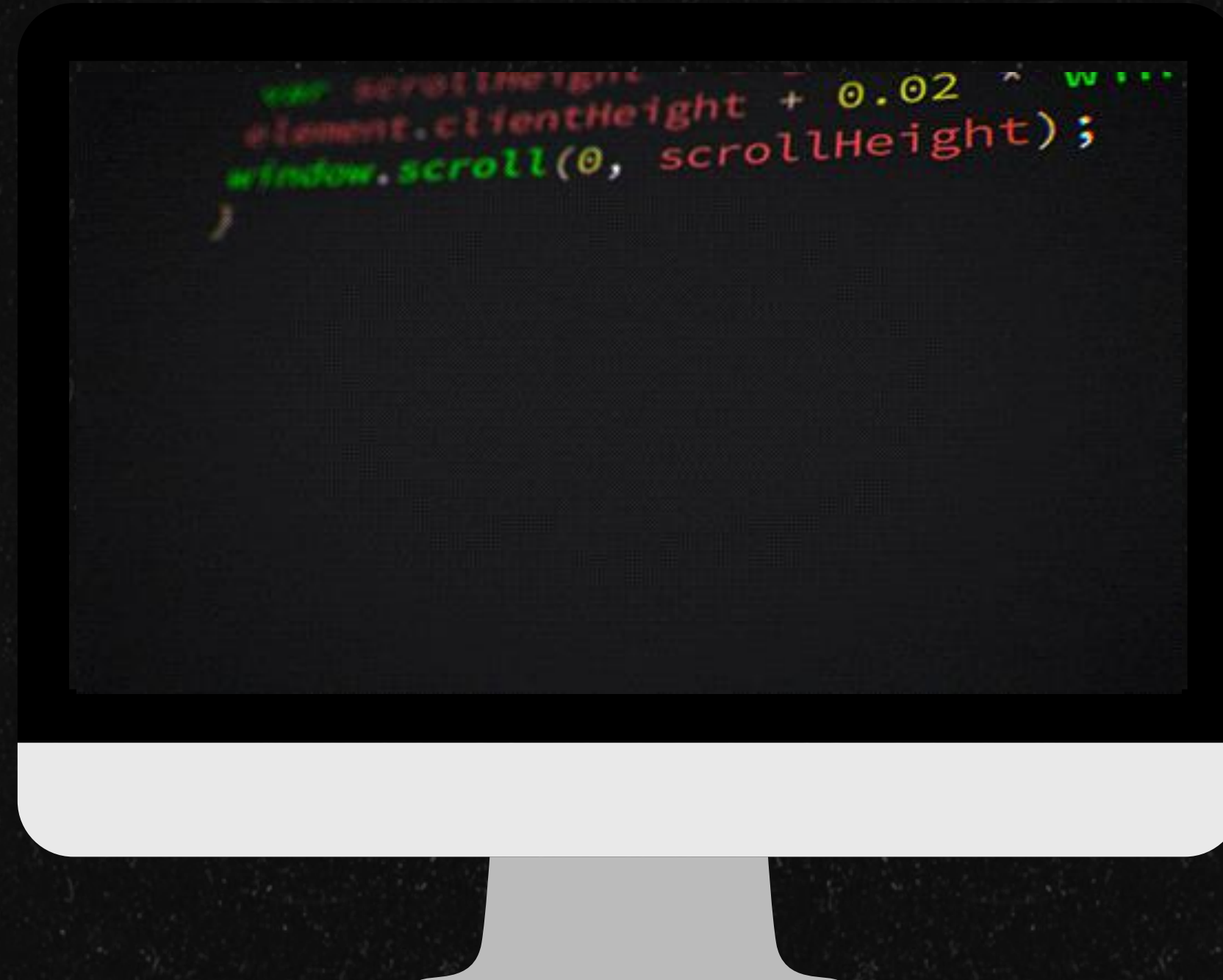
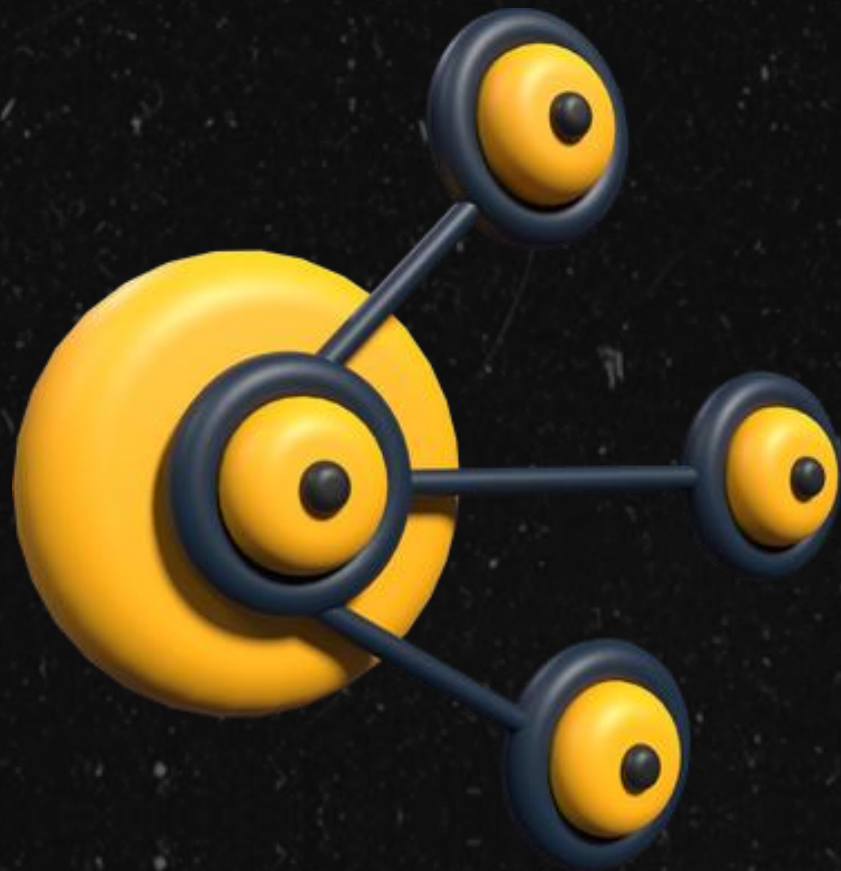
Suyun sıcaklığına göre maddenin katı, sıvı veya gaz olma durumunu gösteren algoritmayı hazır layınız.



```
Adım 1- Başla  
Adım 2- Oku sıcaklık  
Adım 3- Eğer sıcaklık<=0 ise  
         "katı" yaz  
Adım 4- Eğer sıcaklık>0 ve  
         sıcaklık<100 ise "sıvı" yaz  
Adım 5- Eğer sıcaklık >=100 ise  
         "gaz" yaz  
Adım 6- Bitir
```

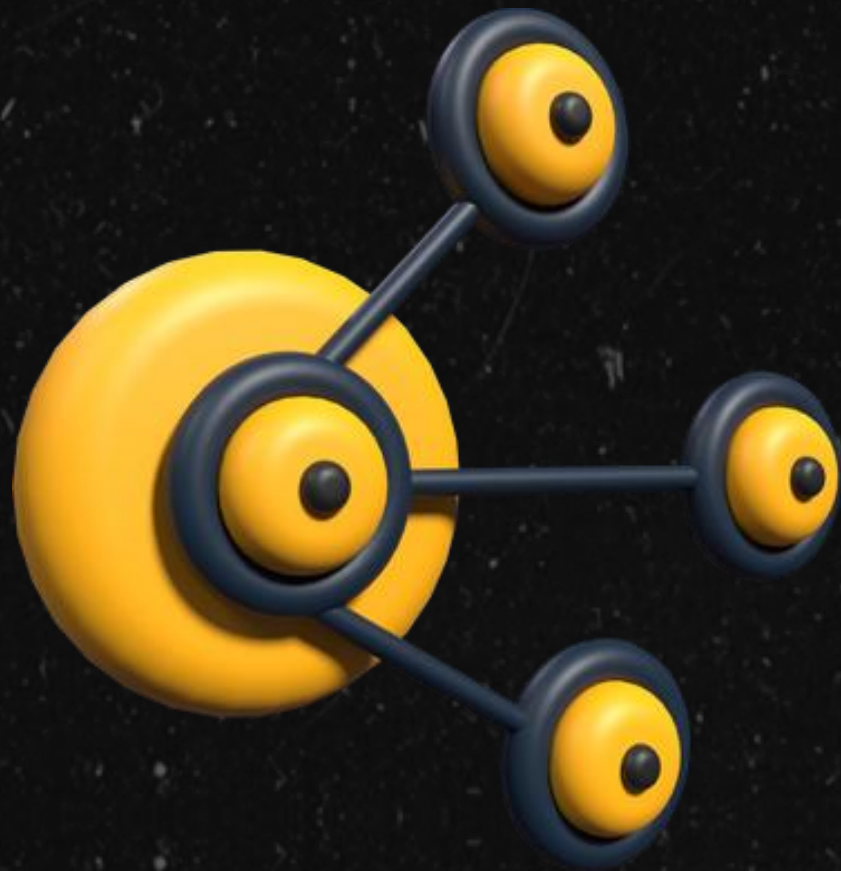
{02} Sahte Kod ve Algoritma

Ekrana 5 defa “merhaba” yazdıran algoritmayı yazınız.



{02} Sahte Kod ve Algoritma

Ekrana 5 defa “merhaba” yazdıran algoritmayı yazınız.



```
Adım 1- Başla  
Adım 2- sayac=0  
Adım 3- yaz “merhaba”  
Adım 4- sayac=sayac+1  
Adım 5- Eğer sayac<5 ise git Adım 3  
Adım 6- Bitir
```

Akış

Diyagramları

```
}  
render() {  
  return (  
    <React.Fragment>  
      <div className="py-5">  
        <div className="container">  
          <Title name="our" title="product">  
            <div className="row">  
              <ProductConsumer>  
                {(value) => {  
                  console.log(value)  
                }}  
              </ProductConsumer>  
            </div>  
          </div>  
        </div>  
      </React.Fragment>  
    )  
  }  
}
```

{03} Akış Diyagramları

Algoritma ile adım adım yapılan işlemlerin, özel semboller (geometrik şekiller) kullanılarak gösterilmesine **akış diyagramı** denir.



{03} Akış Diyagramları

Akış diyagramları görsel olarak basitçe bir problemin akışının gösterimidir. Microsoft firmasının Visio, Word, Powerpoint gibi birçok program ile bu grafikleri hazırlayabilirsiniz.

Akış diyagramındaki standart sembollerle programın mantığı sunulmalıdır.

Akış diyagramı basit, açık ve kolay takip edilebilir olmalıdır.

Her akış diyagramının bir adet başlangıç ve bitiş noktası bulunmalıdır.



{03} Akış Diyagramları

Akış diyagramının takip yönü yukarıdan aşağıya, soldan sağa doğru olmalıdır.

Veri girişleri mantıklı bir sırada listelenmelidir.

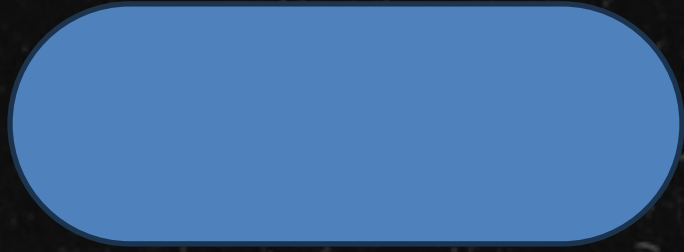
Başlangıç ve bitiş sembollerinde tek hat kullanılmalıdır.

Akış diyagramı çok karmaşık hâle gelince, bağlantı noktaları ve alt program kullanılmalıdır.

Test verisi ile şema denenmelidir. Değerleri bir tabloya aktarılmalıdır.



{03} Akış Diyagramları



Başla-Bitir
Algoritmanın başladığını ve bittiğini gösteren semboldür.



Aritmetik işlemlerin ve değişkene değer atama işlemlerinin yapıldığı semboldür.



Veri giriş ve değişken tanımlamalarının yapıldığı semboldür.



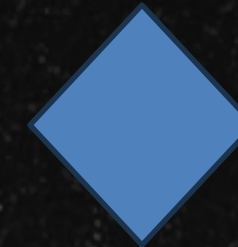
Önceden tanımlı işlem / fonksiyonları çalıştırmak için kullanılan semboldür



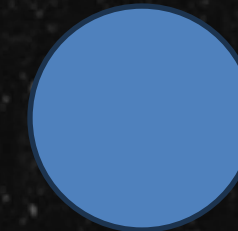
Çıktı almak ve ekran görüntüsü oluşturmak için kullanılan semboldür.



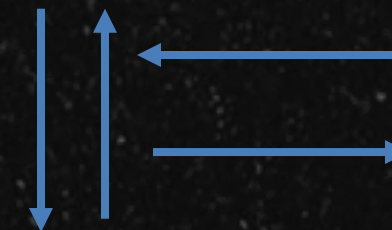
Döngü işlemleri için kullanılan semboldür.



Karar verme / karşılaştırma işlemleri için kullanılan semboldür.



Akış noktalarını bağlamak için kullanılan semboldür.



Akış yönünü gösterir. Akış, okun yönüne göre ilerlemektedir.

Başla - Bitir

İşlem -
Hesaplama

Giriş - Çıkış

Alt program
- Fonksiyon

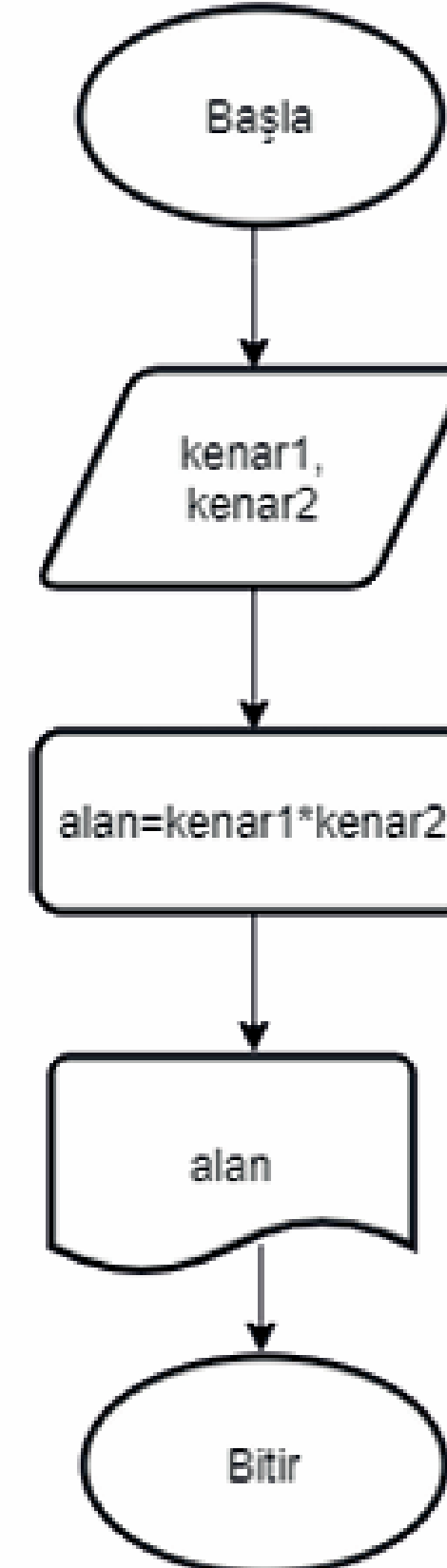
Ekran Çıktısı

Döngü –
Tekrarlı işlemKara
verme –
Karşılaştı
rma –
Koşul

Dikdörtgenin alanını hesaplayan
algoritmayı ve akış şemasını hazırlayınız.

«Sözde Kod»

```
A1-Başla  
A2-Oku kenar1  
A3-Oku kenar2  
A4- $\text{alan} = \text{kenar1} * \text{kenar2}$   
A5-Yaz alan  
A6-Bitir
```



Başla - Bitir

İşlem -
Hesaplama

Giriş - Çıkış

Alt program
- Fonksiyon

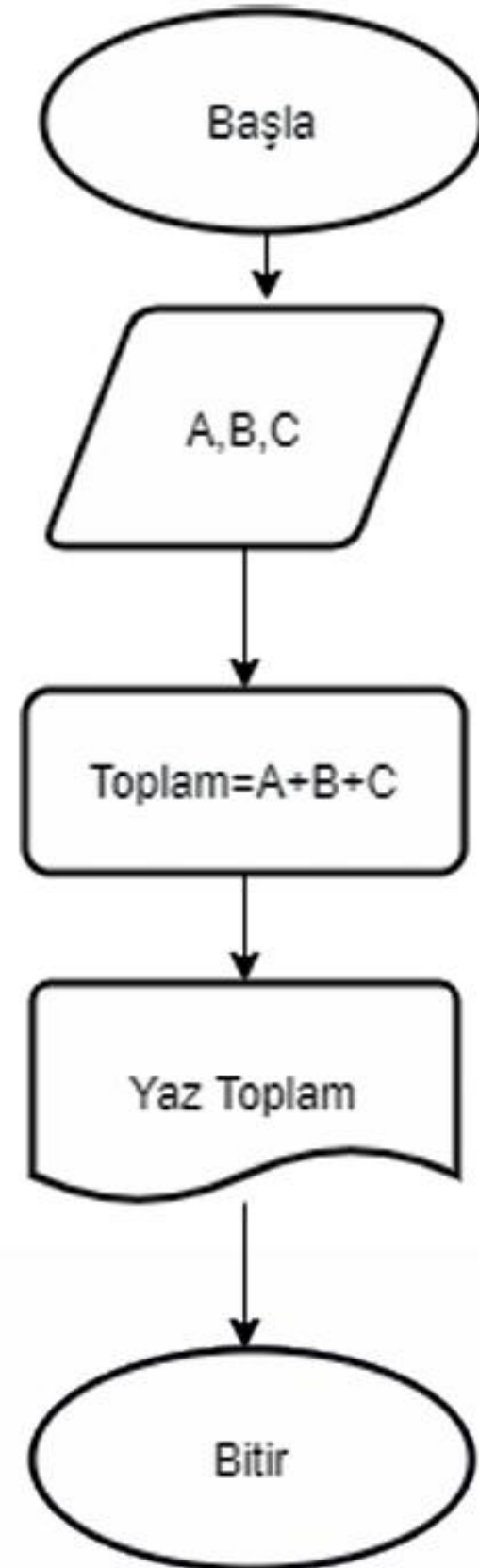
Ekran Çıktısı

Döngü –
Tekrarlı işlemKara
verme –
Karşılaştı
rma –
Koşul

Klavyeden girilecek 3 sayıyı toplayıp sonucu ekrana yazdıran programın algoritmasını ve akış şemasını hazırlayınız.

«Sözde Kod»

```
A1-Başla  
A2-Sayıları giriniz;  
   A,B,C  
A3-Toplam=A+B+C  
A4-Yaz Toplam  
A5-Bitir
```



Başla - Bitir

İşlem -
Hesaplama

Giriş - Çıkış

Alt program
- Fonksiyon

Ekran Çıktısı

Döngü –
Tekrarlı işlemKara
verme –
Karşılaştı
rma –
Koşul

Klavyeden girilen bir sayının negatif mi, pozitif mi yoksa sıfır mı olduğunu yazdıran programın algoritmasını ve akış şemasını hazırlayınız.

«Sözde Kod»

```
A1- Başla  
A2- Sayıyı gir;Sayi  
  
A3- Eğer Sayi<0 ise yaz  
    "girilen sayı negatiftir" ve A6 ya git  
  
A4- Eğer Sayi>0 ise yaz  
    "girilen sayı pozitiftir" ve A6 ya git  
  
A5- Girilen sayı sıfırdır  
A6- Bitir
```

Başla - Bitir

İşlem -
Hesaplama

Giriş - Çıkış

Alt program
- Fonksiyon

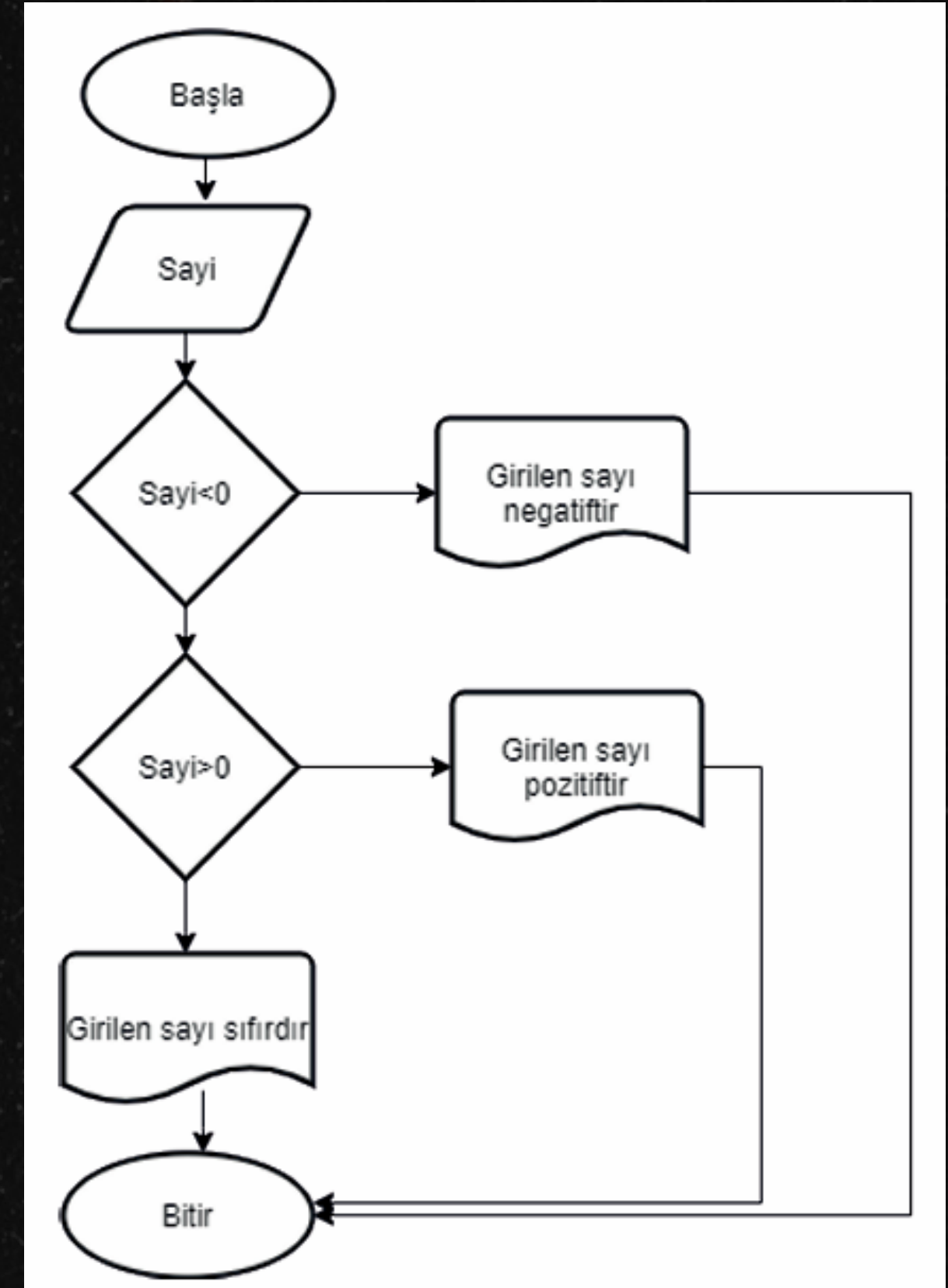
Ekran Çıktısı

Döngü –
Tekrarlı işlemKara
verme –
Karşılaştı
rma –
Koşul

Klavyeden girilen bir sayının negatif mi, pozitif mi yoksa sıfır mı olduğunu yazdıran programın algoritmasını ve akış şemasını hazırlayınız.

«Sözde Kod»

```
A1- Başla  
A2- Sayıyı gir;Sayı  
  
A3- Eğer Sayı<0 ise yaz  
    "girilen sayı negatiftir" ve A6 ya git  
  
A4- Eğer Sayı>0 ise yaz  
    "girilen sayı pozitiftir" ve A6 ya git  
  
A5- Girilen sayı sıfırdır  
A6- Bitir
```



Başla - Bitir

İşlem -
Hesaplama

Giriş - Çıkış

Alt program
- Fonksiyon

Ekran Çıktısı

Döngü –
Tekrarlı işlemKara
verme –
Karşılaştı
rma –
Koşul

Klavyeden girilen iki sayıdan birincisi büyük ise toplama, ikincisi büyük ise çarpma işlemi yapan algoritmayı ve akış şemasını hazırlayınız.

«Sözde Kod»

```
A1-Başla  
A2-Sayıları gir; x,y  
A3-Eğer x>y ise sonuc=x+y ve git A5  
A4-Değilse sonuc=x*y  
A5-Yaz sonuc  
A6-Bitir
```

Başla - Bitir

İşlem -
Hesaplama

Giriş - Çıkış

Alt program
- Fonksiyon

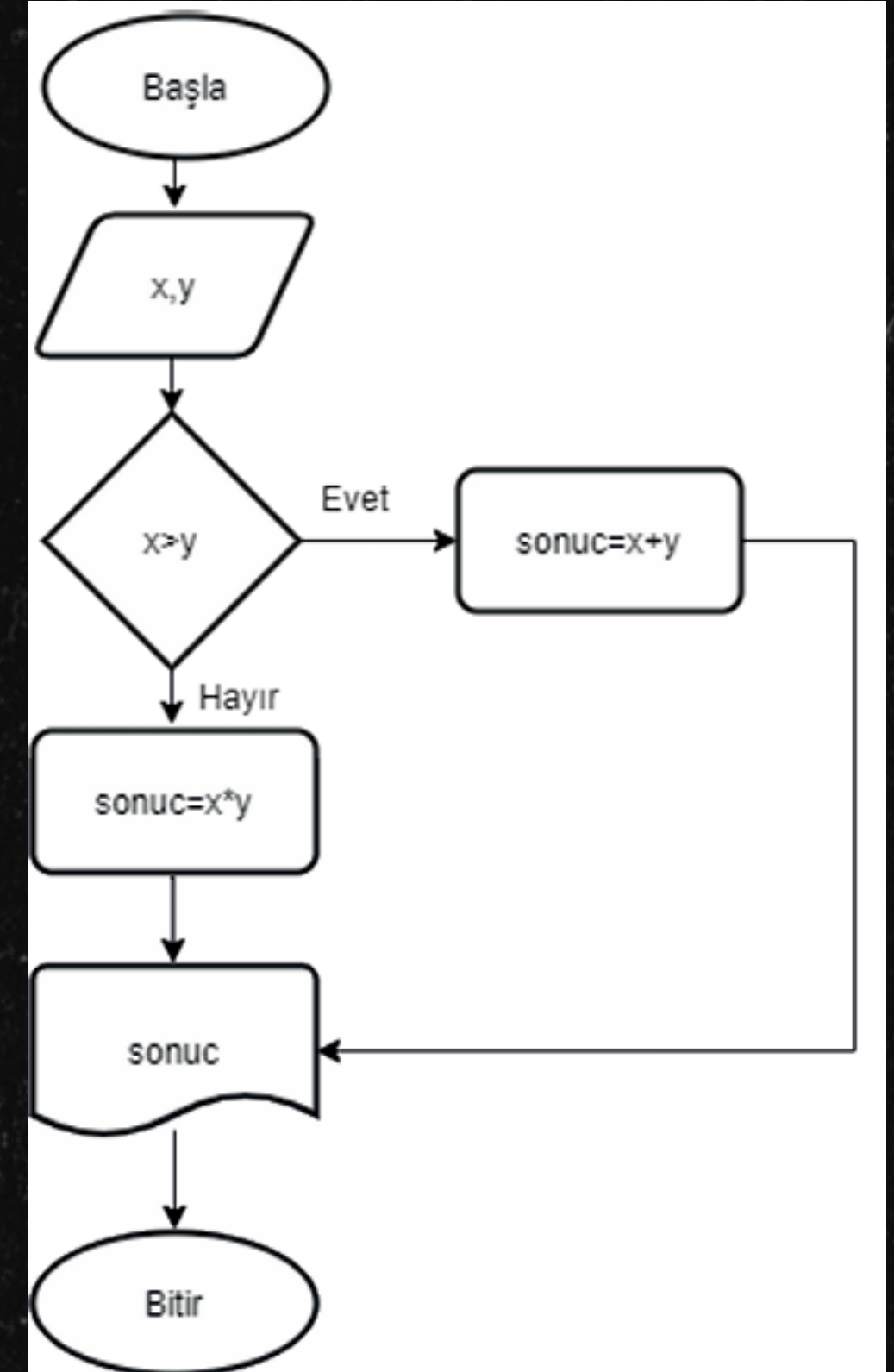
Ekran Çıktısı

Döngü –
Tekrarlı işlemKara
verme –
Karşılaştı
rma –
Koşul

Klavyeden girilen iki sayıdan birincisi büyük ise toplama, ikincisi büyük ise çarpma işlemi yapan algoritmayı ve akış şemasını hazırlayınız.

«Sözde Kod»

```
A1-Başla  
A2-Sayıları gir; x,y  
A3-Eğer x>y ise sonuc=x+y ve git A5  
A4-Değilse sonuc=x*y  
A5-Yaz sonuc  
A6-Bitir
```



Başla - Bitir

İşlem -
Hesaplama

Giriş - Çıkış

Alt program
- Fonksiyon

Ekran Çıktısı

Döngü –
Tekrarlı işlemKara
verme –
Karşılaştı
rma –
Koşul

20 Öğrencinin Temel Programlama 1 dersi birinci sınav notları giriliyor. Geçme notu 60 olan sistemde, kalan öğrenci sayısını bulan algoritmayı ve akış şemasını hazırlayınız.

«Sözde Kod»

```
A1- Başla  
A2- sayac=0, kalan=0  
A3- Sınav Notunu Gir, s_not  
A4- Eğer s_not<60 ise kalan = kalan + 1  
A5- sayac = sayac + 1  
A6- Eğer sayac <= 20 ise, Git A3  
A7- Yaz kalan  
A8- Bitir
```

Başla - Bitir

İşlem -
Hesaplama

Giriş - Çıkış

Alt program
- Fonksiyon

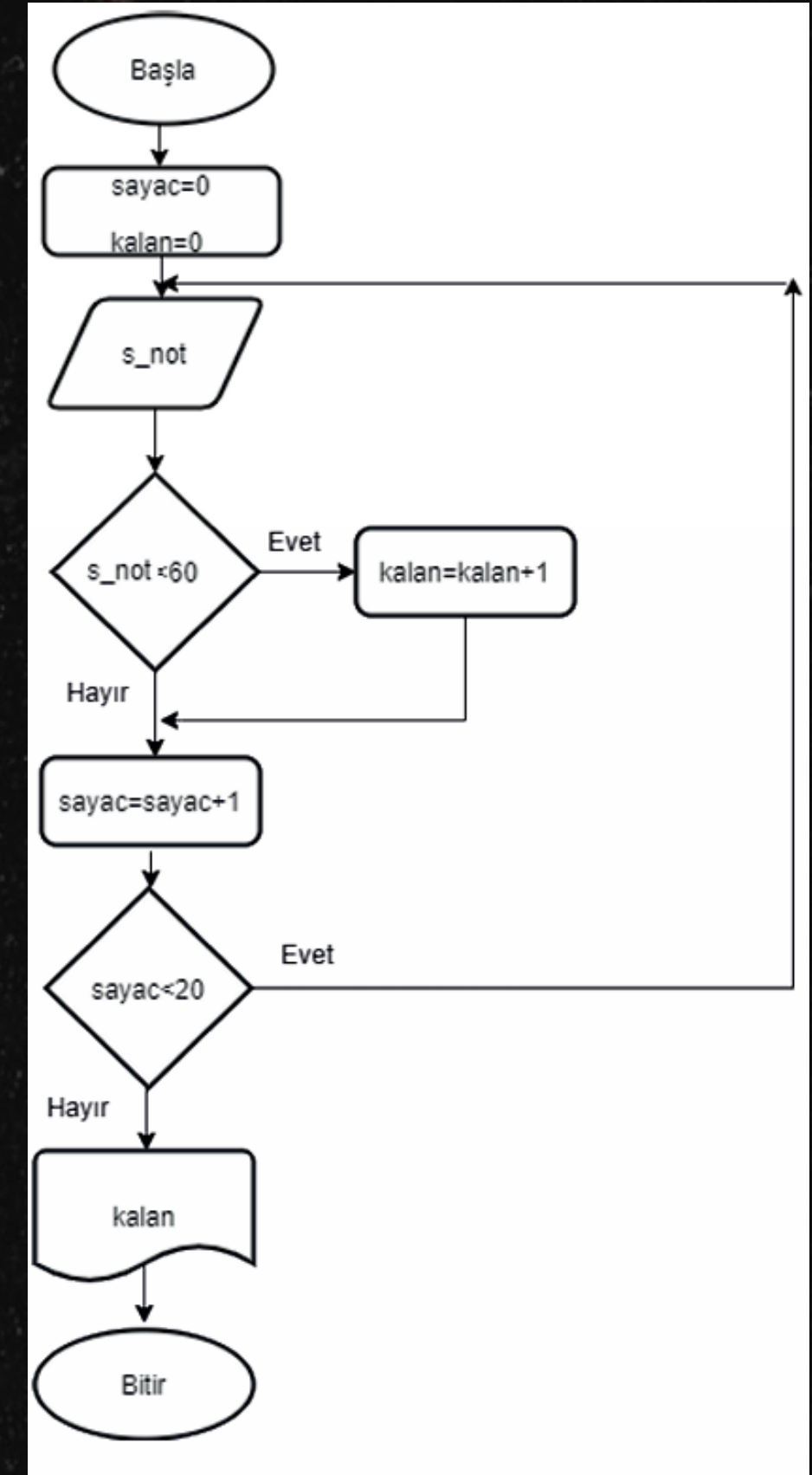
Ekran Çıktısı

Döngü –
Tekrarlı işlemKara
verme –
Karşılaştı
rma –
Koşul

20 Öğrencinin Programlama Temelleri dersi
birinci sınav notları giriliyor. Geçme notu 60
olan sistemde, kalan öğrenci sayısını bulan
algoritmayı ve akış şemasını hazırlayınız.

«Sözde Kod»

```
A1- Başla  
A2- sayac=0, kalan=0  
A3- Sınav Notunu Gir, s_not  
A4- Eğer s_not<60 ise kalan = kalan + 1  
A5- sayac = sayac + 1  
A6- Eğer sayac <= 20 ise, Git A3  
A7- Yaz kalan  
A8- Bitir
```



Başla - Bitir

İşlem -
Hesaplama

Giriş - Çıkış

Alt program
- Fonksiyon

Ekran Çıktısı

Döngü –
Tekrarlı işlemKara
verme –
Karşılaştı
rma –
Koşul

Klavyeden girilen 10 sayının toplamını hesaplayan algoritmayı ve akış diyagramını hazırlayınız.

«Sözde Kod»

```
A1- Başla  
A2- toplam=0,  
A3- Döngü başlat (sayi,1'den 10'a kadar)  
A4- Oku sayi  
A5- toplam=toplam+sayi  
A6- Döngüyü bitir  
A7»- Yaz toplam A7- Bitir
```

Başla - Bitir

İşlem -
Hesaplama

Giriş - Çıkış

Alt program
- Fonksiyon

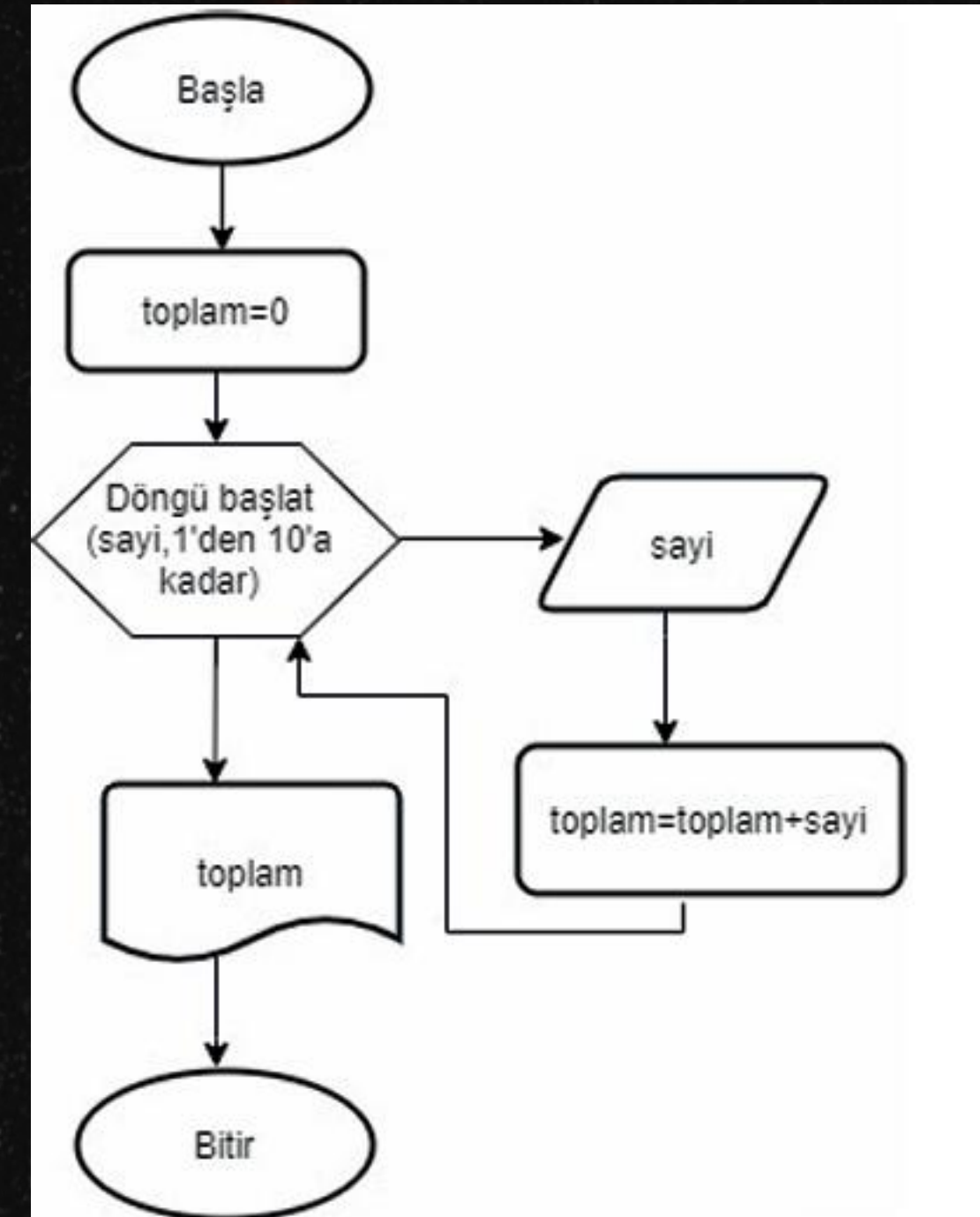
Ekran Çıktısı

Döngü –
Tekrarlı işlemKara
verme –
Karşılaştı
rma –
Koşul

Klavyeden girilen 10 sayının toplamını hesaplayan algoritmayı ve akış diyagramını hazırlayınız.

«Sözde Kod»

A1- Başla
A2- toplam=0,
A3- Döngü başlat (sayi,1'den 10'a kadar)
A4- toplam=toplam+sayi
A5- Döngüyü bitir
A6- Yaz toplam A7- Bitir



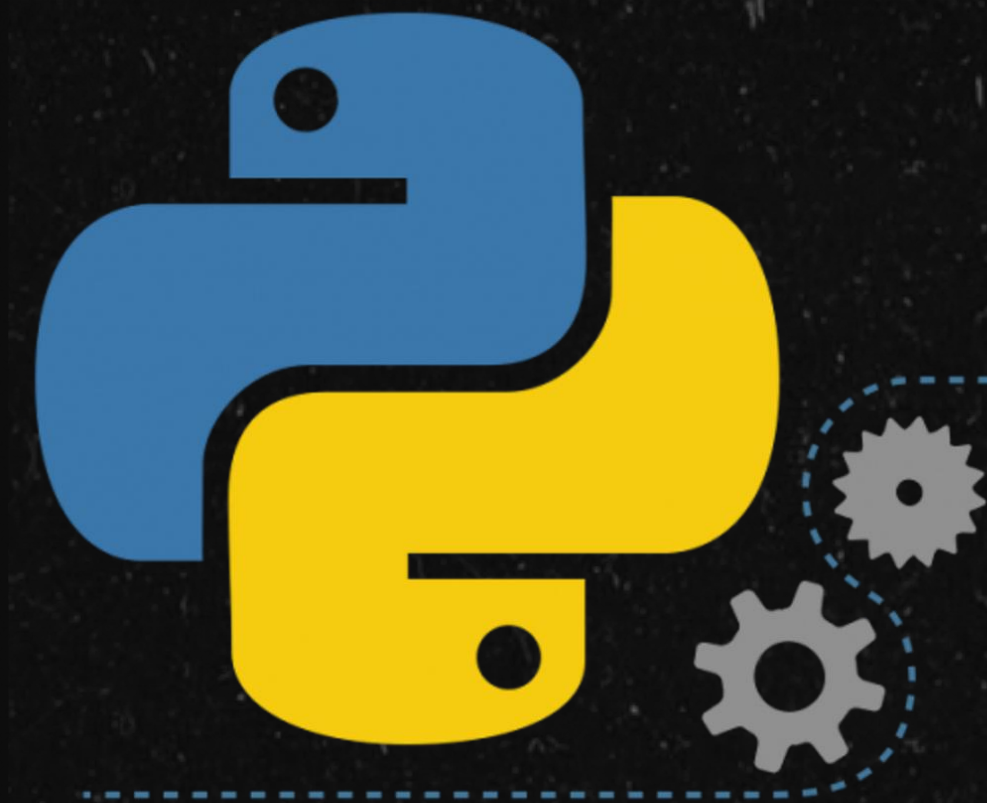
Neden Python

```
response = requests.get(url)
# checking response.status_code (if you get 502, try rerunning the code)
if response.status_code != 200:
    print(f"Status: {response.status_code} - Try rerunning the code\n")
    print(f"Status: {response.status_code}\n")
# using BeautifulSoup to parse the response object
soup = BeautifulSoup(response.content, "html.parser")
# finding Post images in the soup
find_all("img", attrs={"alt": "Post image"})
```

{04} Neden Python

Python programlama dilinin basit ve temiz bir söz dizimi vardır. Bu özelliğinden dolayı program yazmak, yazılan programı okumak ve anlamak diğer dillere nazaran daha kolaydır.

Python'un önemli bir özelliği de pek çok dilin aksine “**yorumlanan**” bir dil olmasıdır. Bu dilde yazılan kodlar derlenmeden direkt çalıştırılır. Python, bu özelliği ile teknik olarak bir programlama dili değil, bir script dilidir. Python'da hızlı bir şekilde program geliştirilebilir. Bu noktada yorumlama ve derleme olaylarını kısaca açıklamak faydalı olacaktır.



{04} Neden Python

Yorumlama (Interpretation) işlemi, yazılan kodun satır satır okunup bilgisayarın işlemcisine özel makine diline anında çevrilmesi işlemidir. Program her çalıştırıldığında yorumlama işlemi tekrardan yapılır. Bundan dolayı derlenen yazılımlara göre bir nebze yavaş çalışacağını söylemek mümkündür.



{04} Neden Python

Derleme (Compilation) işlemi; yüksek seviyeli bir dilde yazılan programın bir başka hedef dile veya makine diline çevrilmesi işlemidir. Programı çalıştırmak için bir kere derleme işleminden geçirmek yeterlidir. Program her çalıştırılışında tekrardan derleme olayı gerçekleşmez.



{04} Neden Python

C++

```
#include <iostream>;  
using namespace std;  
int main(){  
    cout << "Merhaba Dünya!" << endl;  
    return 0;  
}
```



{04} Neden Python

Python:

```
print("Merhaba Dünya!")
```

Her iki program da ekrana “Merhaba Dünya!” yazdırmaktadır. Ancak birinde 6 satır, diğerinde sadece 1 satır kod yazılmıştır. Bu örneklerle bakıldığında Python’un son derece basit ve hızlı bir şekilde kod geliştirilebilen bir programlama dili olduğu görülmektedir.

Python, birçok işletim sisteminde sıkıntısız bir şekilde çalıştırılabilmektedir:
Windows, Linux / Unix, MacOS X ve daha fazlası.

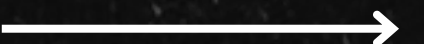


{04} Neden Python



Python'u farklı kılan bazı özellikler şu şekilde listelenebilir:

- Açık kaynak kodlu olması
- Ücretsiz olması
- Hızlı ve kolay kurulabilmesi
- Sade ve kolay okunabilen kod yapısı
- Toplu ve düzenli kod yapısı
- Öğrenme ve adapte olma kolaylığı
- Kolay anlaşılır nesne tabanlı programlama özellikleri
- Güçlü ifade yeteneği
- Son derece esnek modüler yapısı
- "Exception" tabanlı hata yönetimi
- Yüksek seviye dinamik veri yapıları
- Oldukça geniş standart kütüphanelerinin olması
- Otomatik hafıza temizliği
- C, C++, Java ile kolay entegre edilebilmesi
- Hemen her tür platformda sıkıntısız çalışması
- Az kod / çok iş anlayışı



{04} Neden Python



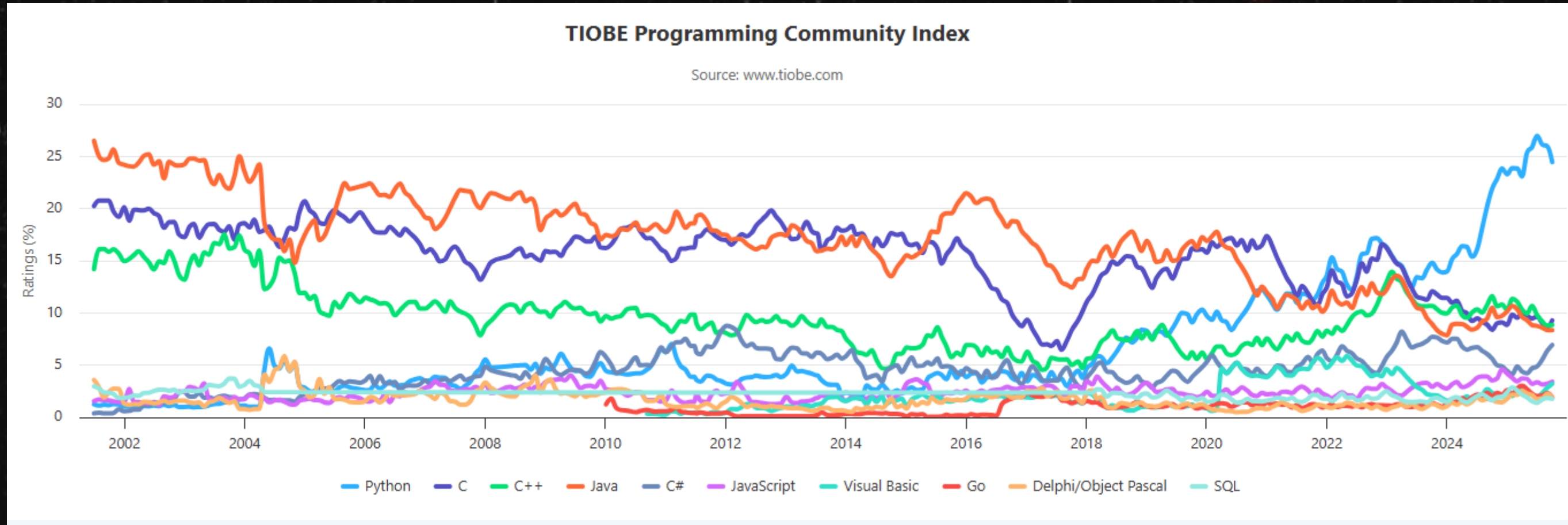
Oct 2025	Oct 2024	Change	Programming Language		Ratings	Change
1	1			Python	24.45%	+2.55%
2	4	▲		C	9.29%	+0.91%
3	2	▼		C++	8.84%	-2.77%
4	3	▼		Java	8.35%	-2.15%
5	5			C#	6.94%	+1.32%
6	6			JavaScript	3.41%	-0.13%
7	7			Visual Basic	3.22%	+0.87%
8	8			Go	1.92%	-0.10%
9	10	▲		Delphi/Object Pascal	1.86%	+0.19%
10	11	▲		SQL	1.77%	+0.13%
11	9	▼		Fortran	1.70%	-0.10%
12	29	▲▲		Perl	1.66%	+1.10%
13	17	▲▲		R	1.52%	+0.43%

{04} Neden Python

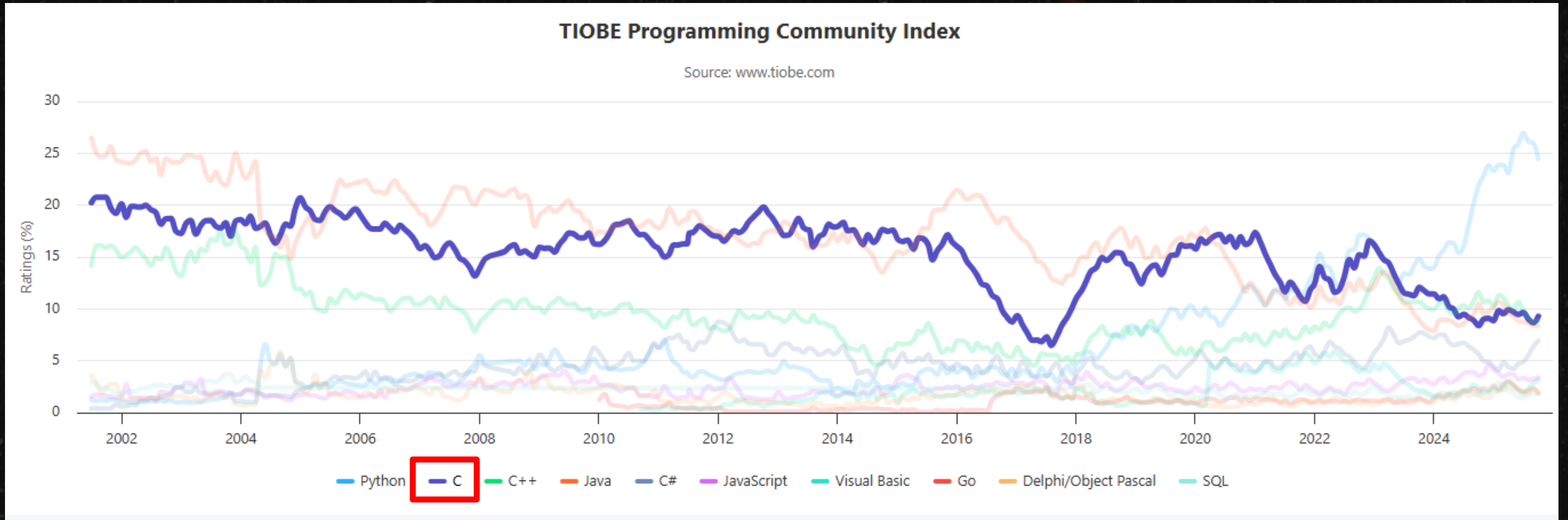


Oct 2025	Oct 2024	Change	Programming Language		Ratings	Change
1	1			Python	24.45%	+2.55%
2	4	▲		C	9.29%	+0.91%
3	2	▼		C++	8.84%	-2.77%
4	3	▼		Java	8.35%	-2.15%
5	5			C#	6.94%	+1.32%
6	6			JavaScript	3.41%	-0.13%
7	7			Visual Basic	3.22%	+0.87%
8	8			Go	1.92%	-0.10%
9	10	▲		Delphi/Object Pascal	1.86%	+0.19%
10	11	▲		SQL	1.77%	+0.13%
11	9	▼		Fortran	1.70%	-0.10%
12	29	▲▲		Perl	1.66%	+1.10%
13	17	▲▲		R	1.52%	+0.43%

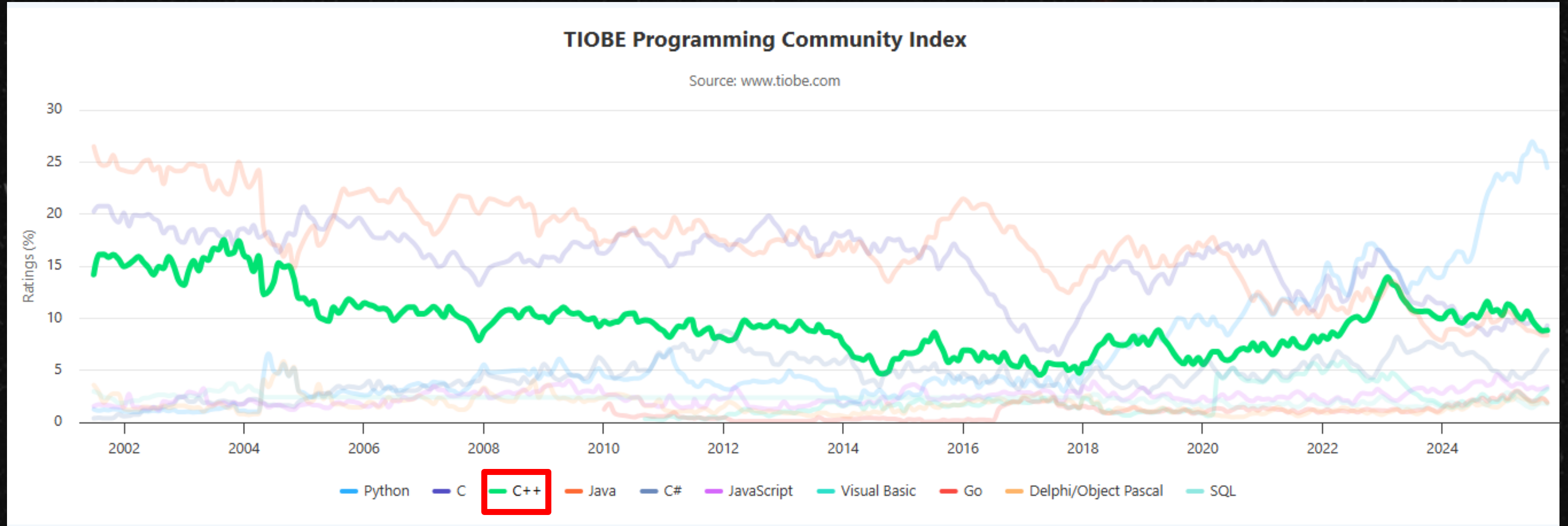
{04} Neden Python



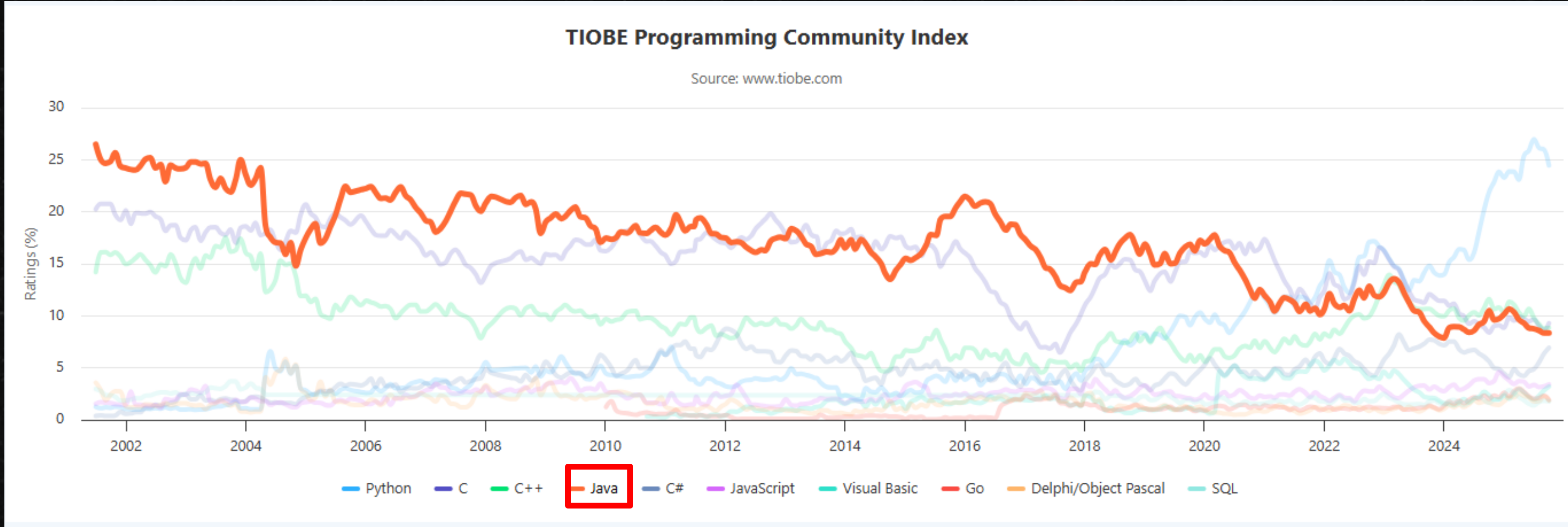
{04} Neden Python



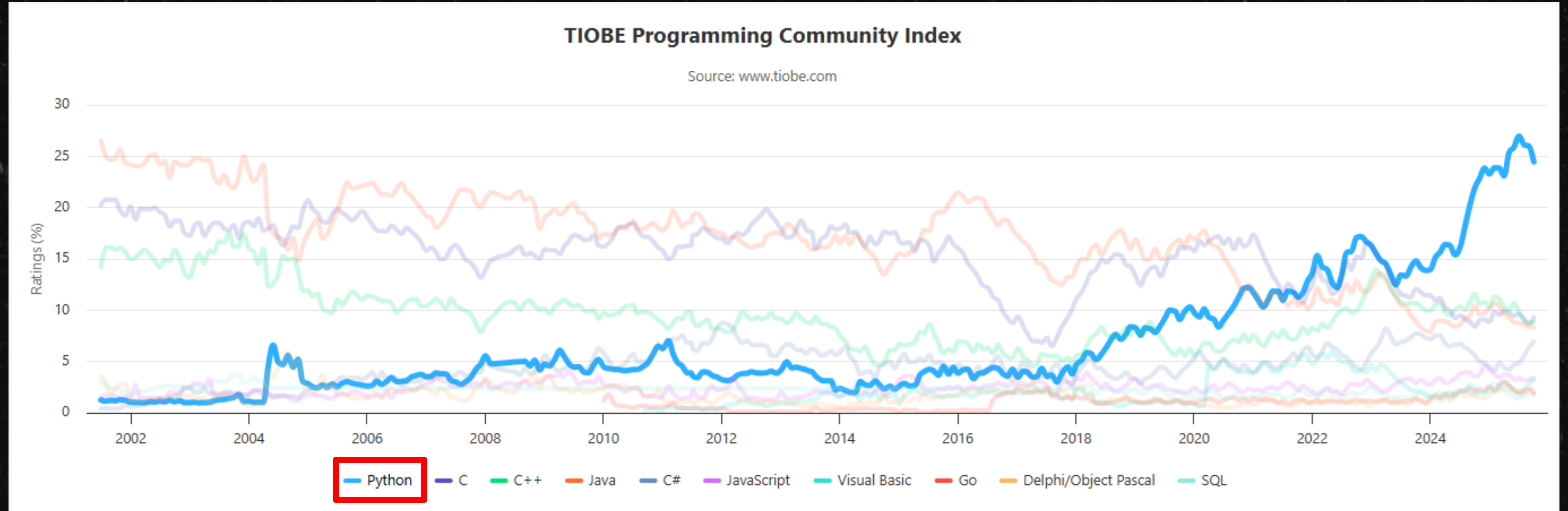
{04} Neden Python



{04} Neden Python



{04} Neden Python



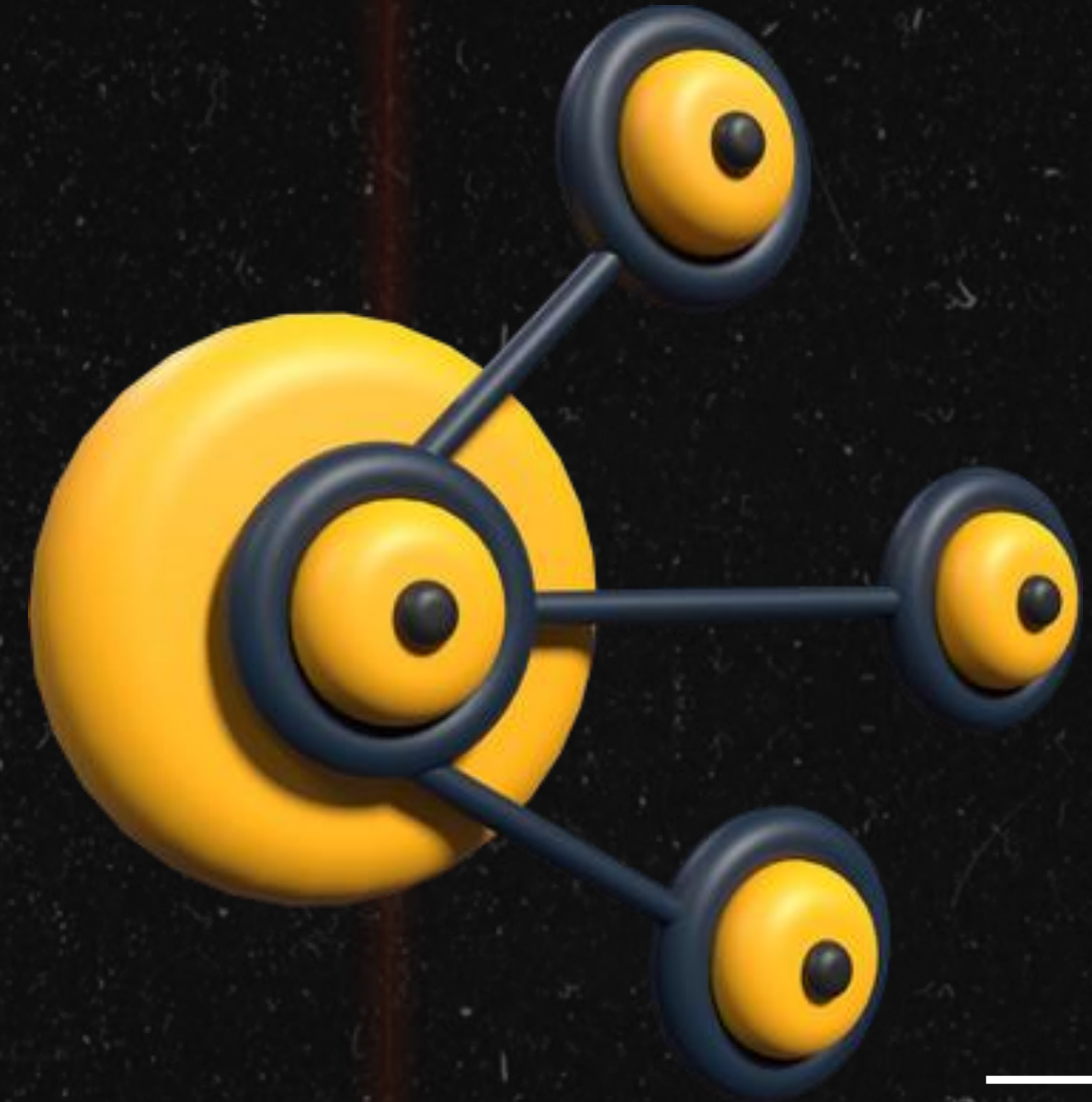
{04} Neden Python

Programming Language	2025	2020	2015	2010	2005	2000	1995	1990	1985
Python	1	3	6	7	7	24	16	-	-
C++	2	4	3	3	3	2	1	2	10
C	3	1	2	2	1	1	2	1	1
Java	4	2	1	1	2	3	32	-	-
C#	5	5	4	6	8	10	-	-	-
JavaScript	6	7	8	9	11	8	-	-	-
Go	7	13	61	172	-	-	-	-	-
Visual Basic	8	10	11	-	-	-	-	-	-
Delphi/Object Pascal	9	188	12	10	9	-	-	-	-
SQL	10	9	-	-	-	-	-	-	-
Ada	17	35	30	24	18	20	5	8	3
Lisp	26	31	28	16	14	9	6	7	2
(Visual) Basic	-	-	-	5	6	4	3	3	4

{04} Neden Python

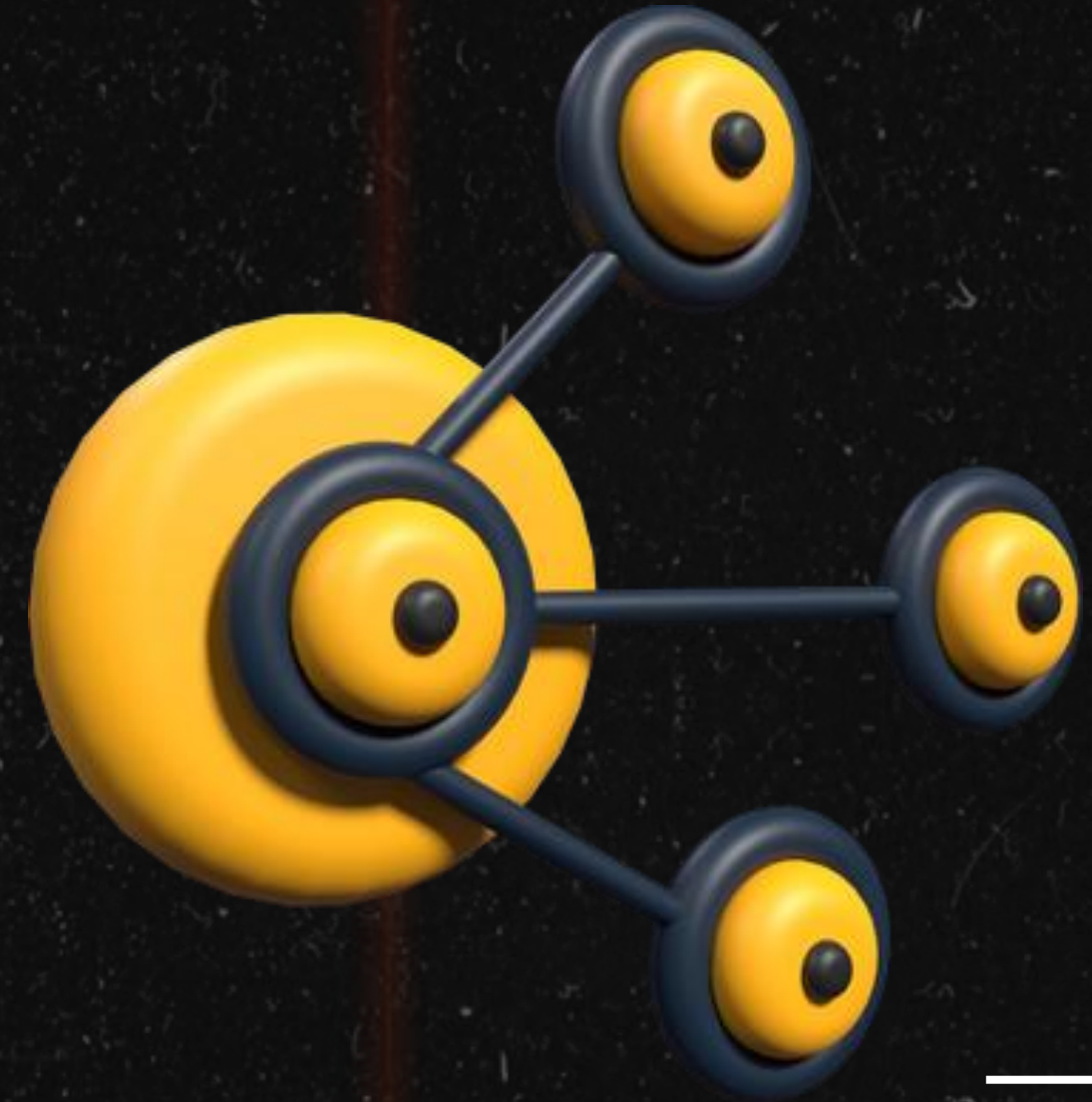
Aşağıda Python dilinin kullanıldığı alanlar listelenmiştir. Bu liste Python'un neden bu kadar popüler olduğunu açıklamaktadır.

- Yapay zekâ ve makine öğrenmesi
- Web uygulamaları
- Bilimsel hesaplamalar
- Veri analizi
- Masaüstü uygulama geliştirme
- Ağ ve soket programlama
- Nesnelerin interneti
- Kriptoloji
- Sistem yönetimi
- Oyun geliştirme vb.



{05}Python Kurulumu

Bunun için Python resmî web sitesi olan python.org adresi ziyaret edilmelidir.



{05} Python Kurulumu

The screenshot shows the Python.org website. The browser address bar displays 'python.org'. The navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this, there's a search bar and a 'Socialize' button. The main content area features a code snippet for simple arithmetic, a section titled 'Intuitive Interpretation' explaining Python's syntax, and a 'Download' button. A red circle highlights the 'Download' button in the footer, which is also circled in a larger red circle in a callout box. The callout box contains the text: 'Download Python source code and installers are available for download for all versions! Latest: Python 3.14.0'. The footer also includes links for 'Get Started', 'Docs', and 'Jobs'.

python.org

Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3: Simple arithmetic
>>> 1 / 2
0.5
>>> 2 ** 3
8
>>> 17 / 3 # classic division returns a float
5.666666666666667
>>> 17 // 3 # floor division
5
```

Intuitive Interpretation

Calculations are simple with Python, and expression syntax is straightforward: the operators `+`, `-`, `*` and `/` work as expected; parentheses `()` can be used for grouping. [More about simple math functions in Python 3.](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn more](#)

Get Started
Whether you're new to programming or an experienced developer, it's easy to learn and use Python.
[Start with our Beginner's Guide](#)

Download
Python source code and installers are available for download for all versions!
Latest: Python 3.14.0

Docs
Documentation for Python's standard library, along with tutorials and guides, are available online.
[docs.python.org](#)

Jobs
Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.
[jobs.python.org](#)

Download
Python source code and installers are available for download for all versions!
Latest: Python 3.14.0

{05} Python Kurulumu

python.org/downloads/release/python-3140/

isation contributions to the [Python Software Foundation](#).

[Full Changelog](#)

Files

Version	Operating System	Description	MD5 Sum	File Size	Sigstore	SBOM
Gzipped source tarball	Source release		2ba6baae1e7c56f652195327d3becd64	29.2 MB	.sigstore	SPDX
XZ compressed source tarball	Source release		41389edaf9c643263cbcd9b5ed307df8	22.5 MB	.sigstore	SPDX
Android embeddable package (aarch64)	Android		be1f8e3ec0752f5bbaf6e029e0a870c0	19.9 MB	.sigstore	
Android embeddable package (x86_64)	Android		c9795011cd212f75303bb4fac18e6f0d	20.0 MB	.sigstore	
macOS 64-bit universal2 installer	macOS	for macOS 10.15 and later	da9338ba3e64cd80f9cc890280933e6c	70.9 MB	.sigstore	
Windows installer (64-bit)	Windows	Recommended	cf642108b97545a30ac055b94657a0e6	28.5 MB	.sigstore	SPDX
Windows installer (32-bit)	Windows		4b4a8f4e6f47a430df110868b3872762	27.1 MB	.sigstore	SPDX
Windows installer (ARM64)	Windows	Experimental	1d308e03a711d99ad7c5df4032daf8a9	27.8 MB	.sigstore	SPDX
Windows embeddable package (64-bit)	Windows		7c5d8d8e3213a11bd0e36f8b8eb03431	11.4 MB	.sigstore	SPDX
Windows embeddable package (32-bit)	Windows		c21d08efbf4d7adb64505577c197bd08	10.1 MB	.sigstore	SPDX
Windows embeddable package (ARM64)	Windows		e207670ef0d54dca3681c9e4137a4774	10.8 MB	.sigstore	SPDX
Windows release manifest	Windows	Install with 'py install 3.14'	1f8f0ad1876492476cbc02155976dc21	14.6 KB	.sigstore	

[About](#) [Downloads](#) [Documentation](#) [Community](#) [Success Stories](#) [News](#)

[Applications](#) [All releases](#) [Docs](#) [Diversity](#) [Arts](#) [Python News](#)

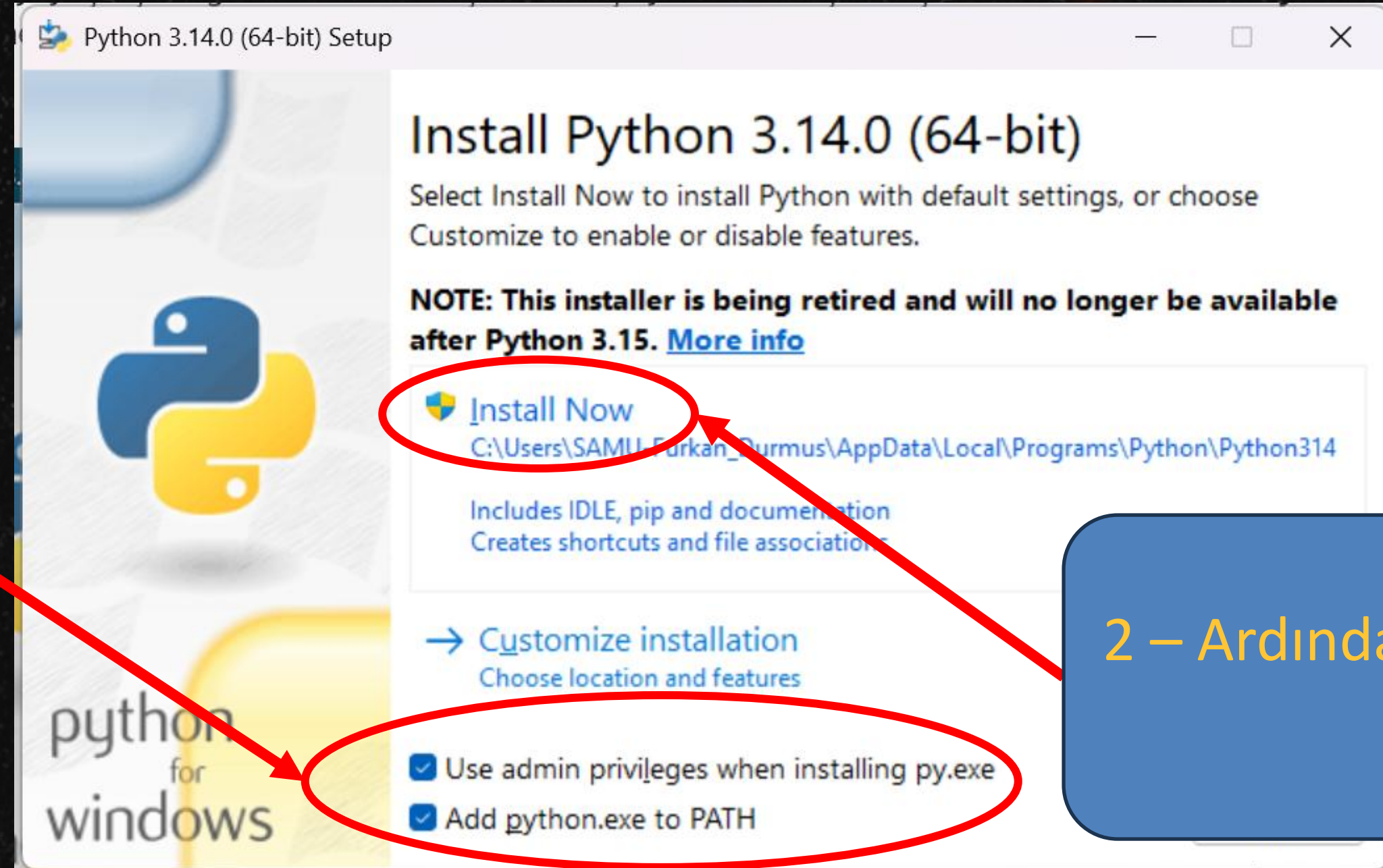
[Source code](#) [Audio/Visual Talks](#) [Mailing Lists](#) [Business](#) [PSF Newsletter](#)

<https://www.python.org/ftp/python/3.14.0/python-3.14.0-amd64.exe>

[macOS 64-bit universal2 installer](#)[Windows installer \(64-bit\)](#)[Windows installer \(32-bit\)](#)

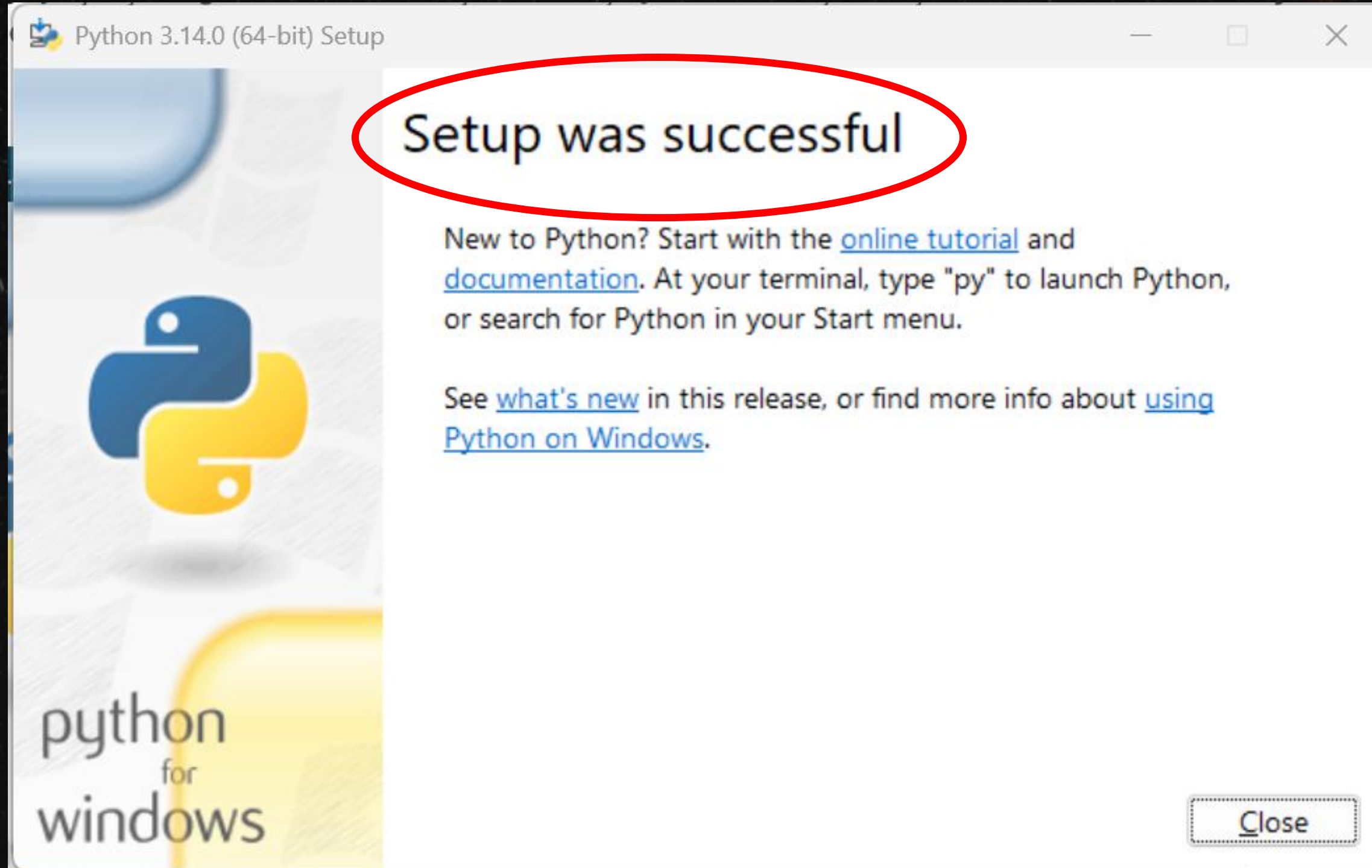
{05} Python Kurulumu

1 -HER İKİ SEÇENEĞİ
İŞARETLE



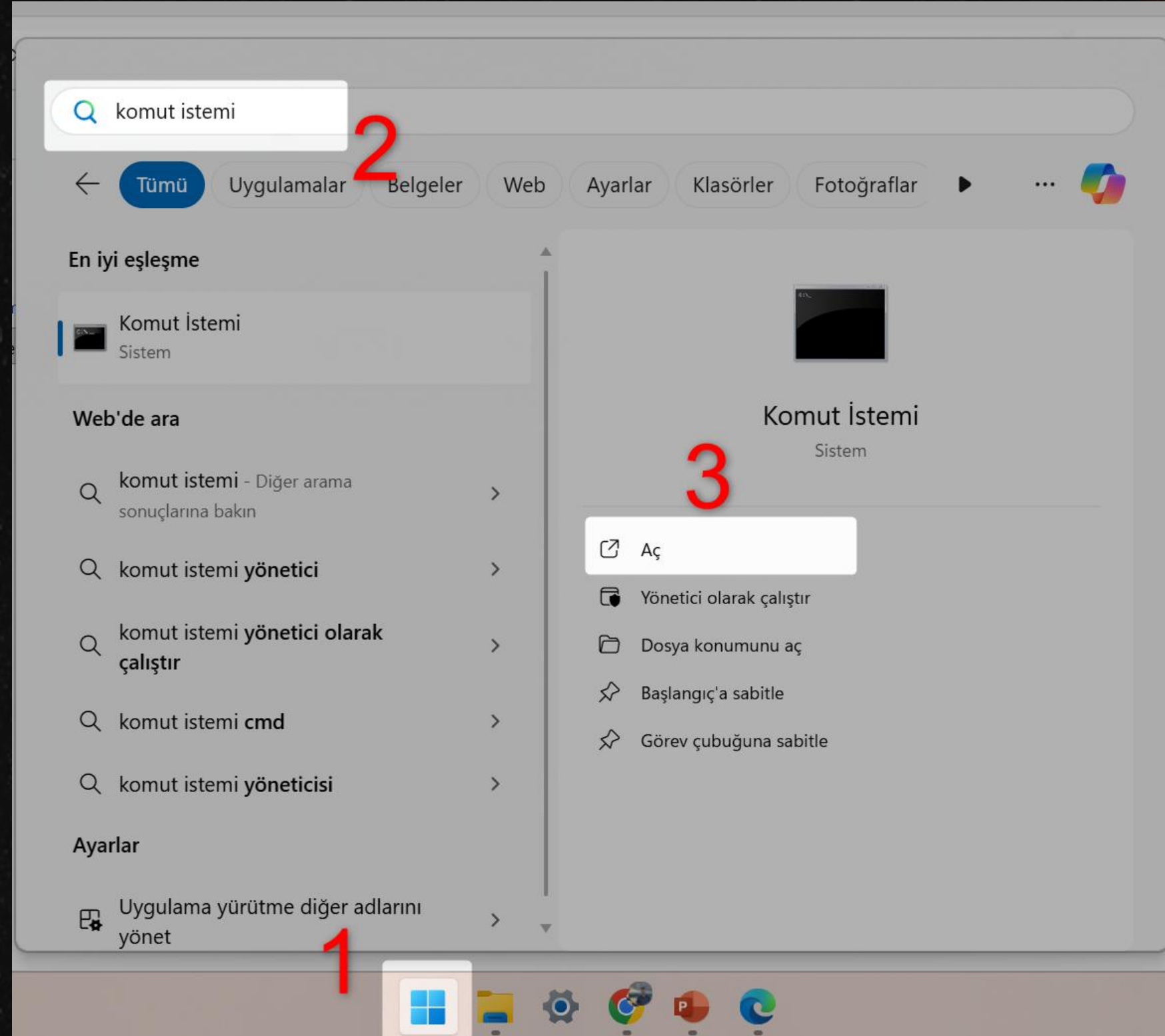
2 – Ardından 'Install Now'
Tıkla

{05} Python Kurulumu

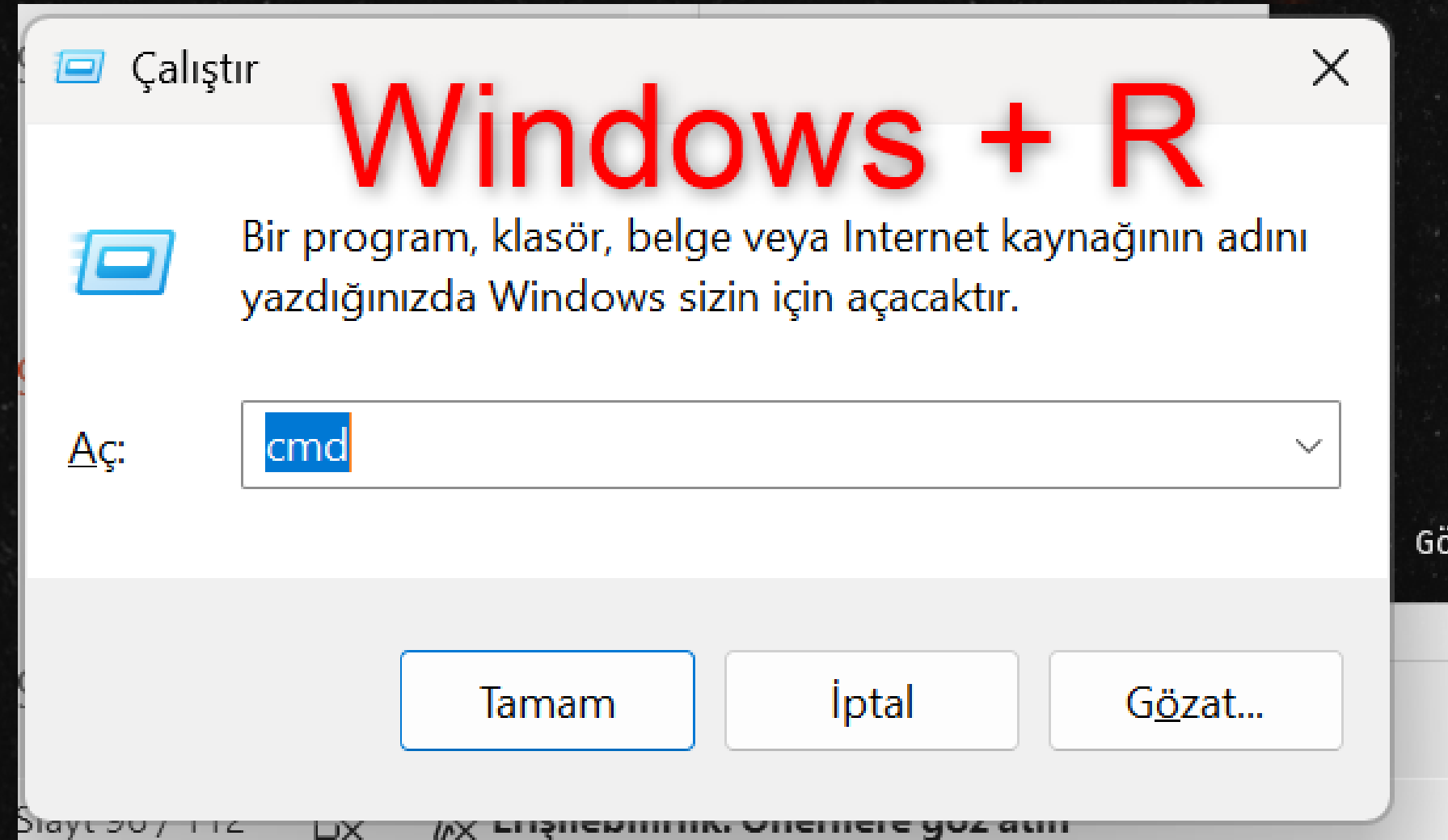


Kurulum Başarılı

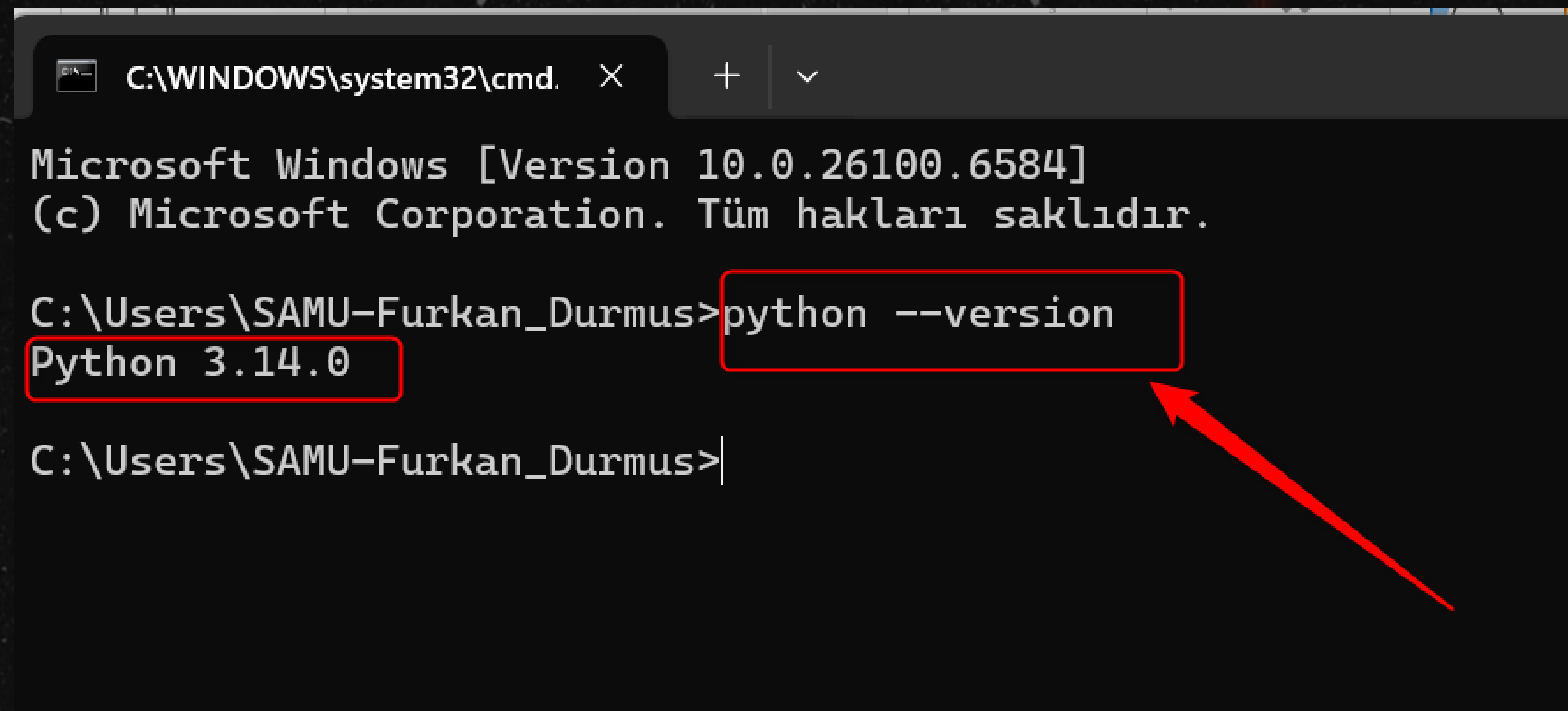
{05} Python Kurulumu



{05}Python Kurulumu



{05}Python Kurulumu



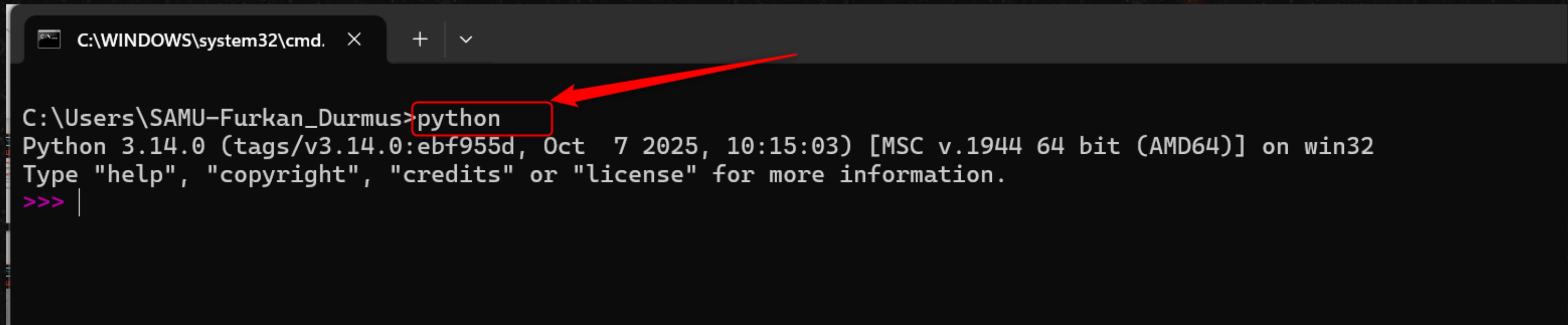
```
C:\WINDOWS\system32\cmd. X + v

Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\SAMU-Furkan_Durmus>python --version
Python 3.14.0

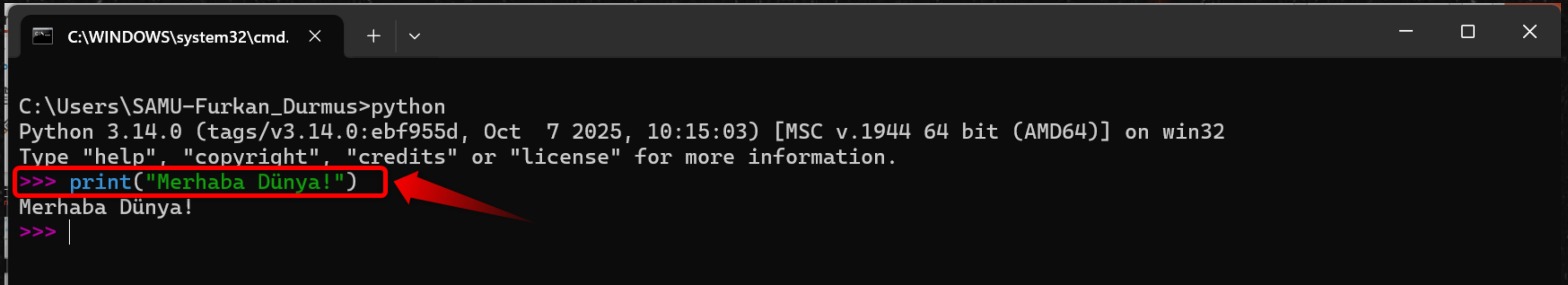
C:\Users\SAMU-Furkan_Durmus>
```

{05}Python Kurulumu



```
C:\WINDOWS\system32\cmd. ✕ + ▾  
C:\Users\SAMU-Furkan_Durmus>python  
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>> |
```

{05}Python Kurulumu



The screenshot shows a Windows Command Prompt window with the title bar 'C:\WINDOWS\system32\cmd.' and standard window controls. The command prompt displays the following text:

```
C:\Users\SAMU-Furkan_Durmus>python
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Merhaba Dünya!")
Merhaba Dünya!
>>> |
```

The command `print("Merhaba Dünya!")` is highlighted with a red box, and a red arrow points to it from the right.

{05}Python Kurulumu

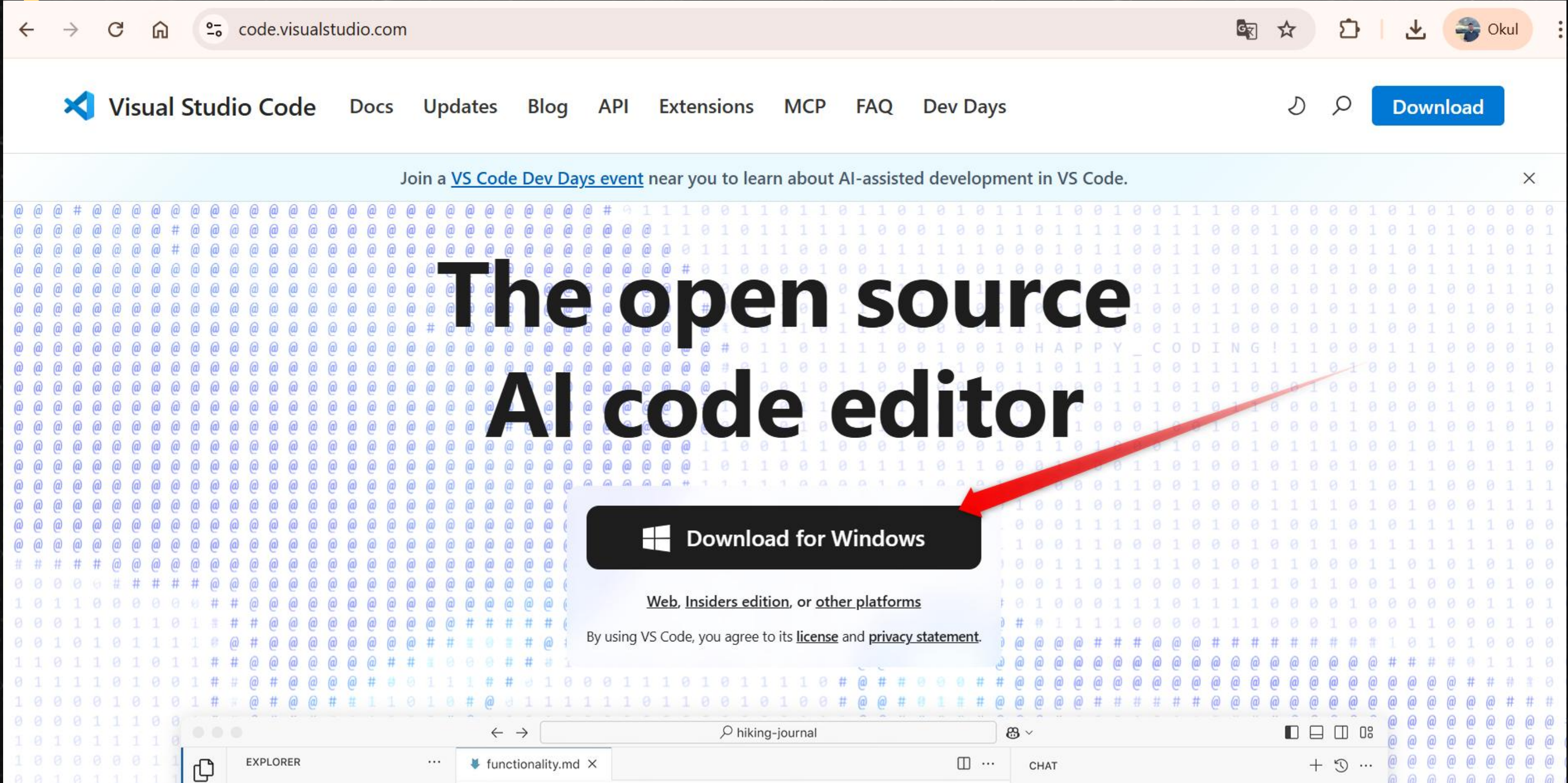
```
C:\WINDOWS\system32\cmd. ✕ + ▾  
  
C:\Users\SAMU-Furkan_Durmus>python  
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 3 + 5  
8  
>>> |
```



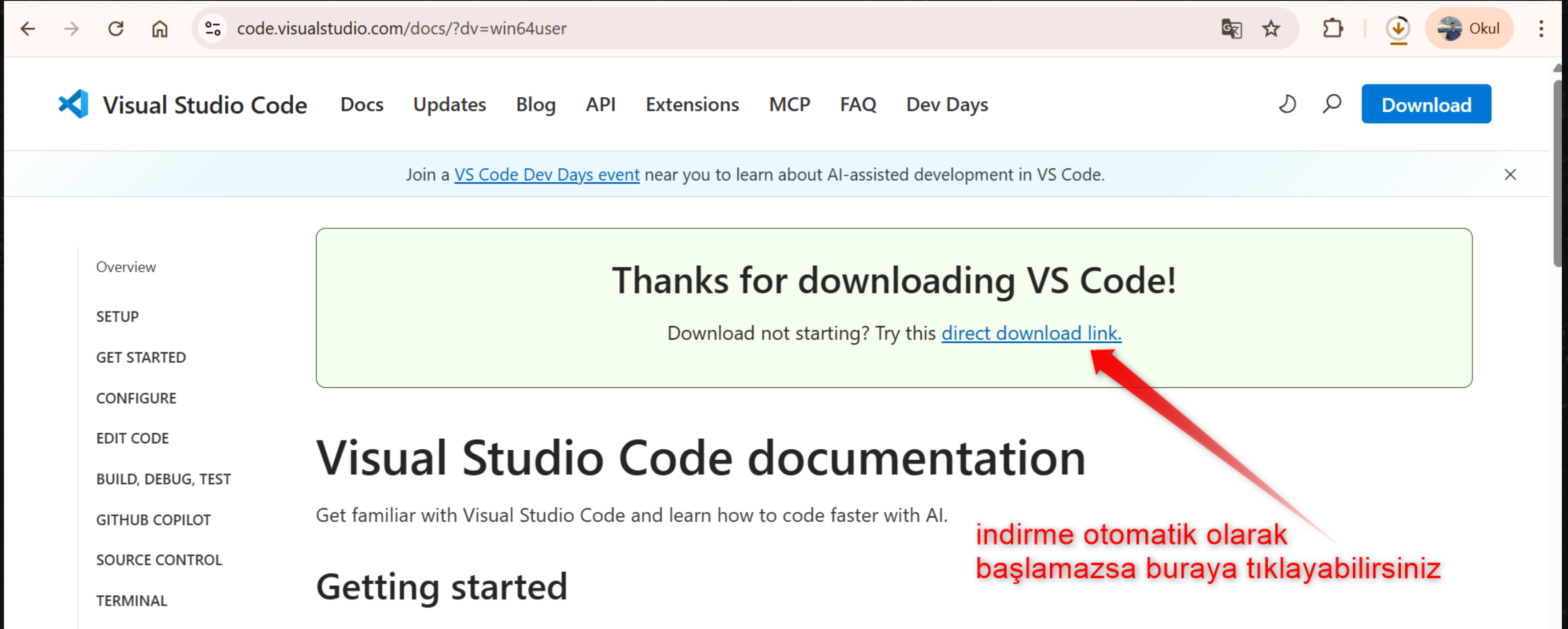
{05} IDE Kurulumu

<https://code.visualstudio.com/>

{05} IDE Kurulumu



{05} IDE Kurulumu



The screenshot shows the Visual Studio Code documentation page. The browser address bar displays 'code.visualstudio.com/docs?dv=win64user'. The page header includes the Visual Studio Code logo and navigation links: Docs, Updates, Blog, API, Extensions, MCP, FAQ, and Dev Days. A blue 'Download' button is visible in the top right. A light blue banner below the header promotes 'VS Code Dev Days event'. The main content area features a green box with the text 'Thanks for downloading VS Code!' and a link 'direct download link.' which is highlighted by a red arrow. Below this, the heading 'Visual Studio Code documentation' is followed by the text 'Get familiar with Visual Studio Code and learn how to code faster with AI.' and the section 'Getting started'. A sidebar on the left lists various topics: Overview, SETUP, GET STARTED, CONFIGURE, EDIT CODE, BUILD, DEBUG, TEST, GITHUB COPILOT, SOURCE CONTROL, and TERMINAL. A red text annotation at the bottom right reads 'indirme otomatik olarak başlamazsa buraya tıklayabilirsiniz'.

code.visualstudio.com/docs?dv=win64user

Visual Studio Code Docs Updates Blog API Extensions MCP FAQ Dev Days

Download

Join a [VS Code Dev Days event](#) near you to learn about AI-assisted development in VS Code.

Thanks for downloading VS Code!

Download not starting? Try this [direct download link.](#)

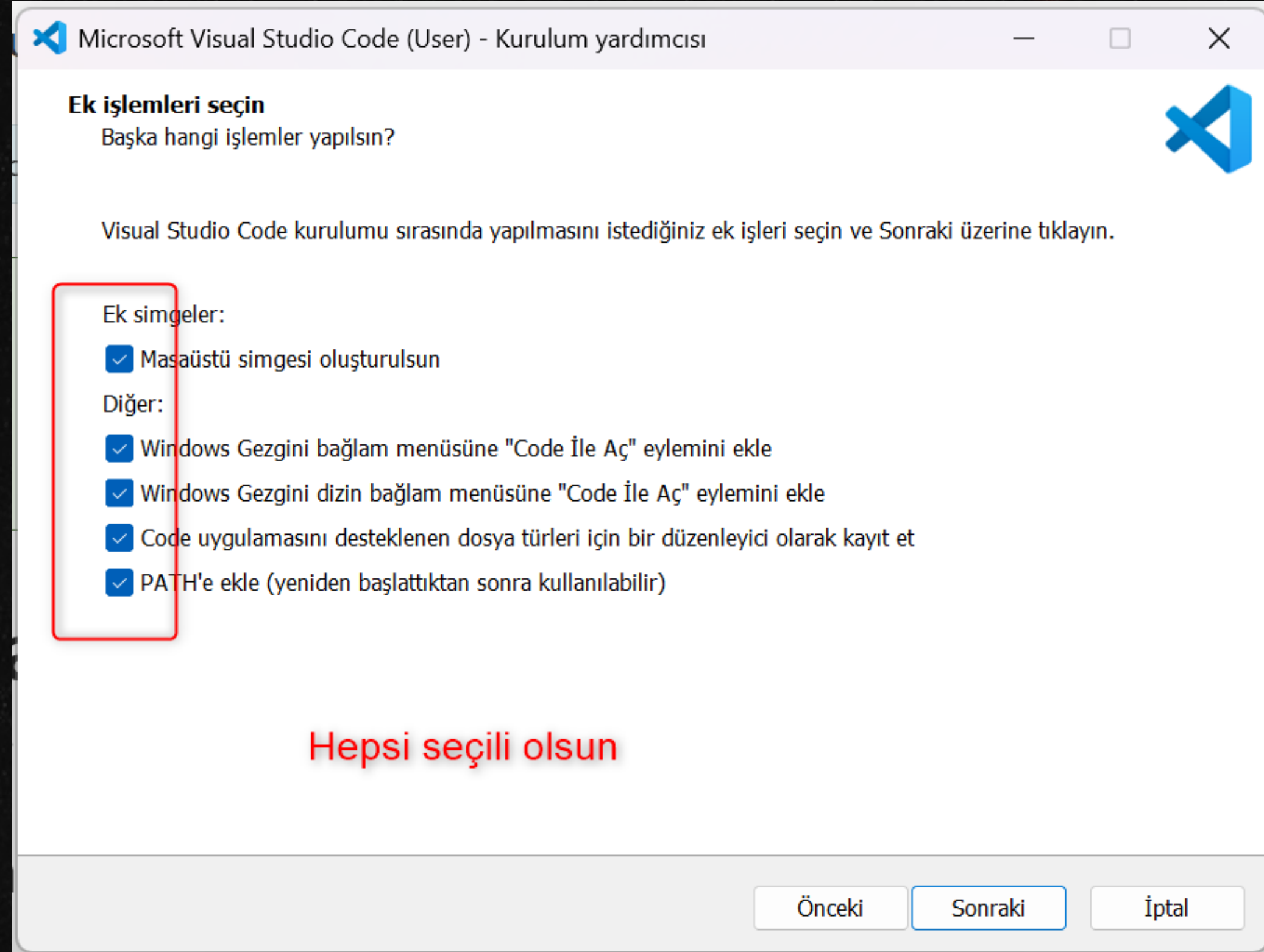
Visual Studio Code documentation

Get familiar with Visual Studio Code and learn how to code faster with AI.

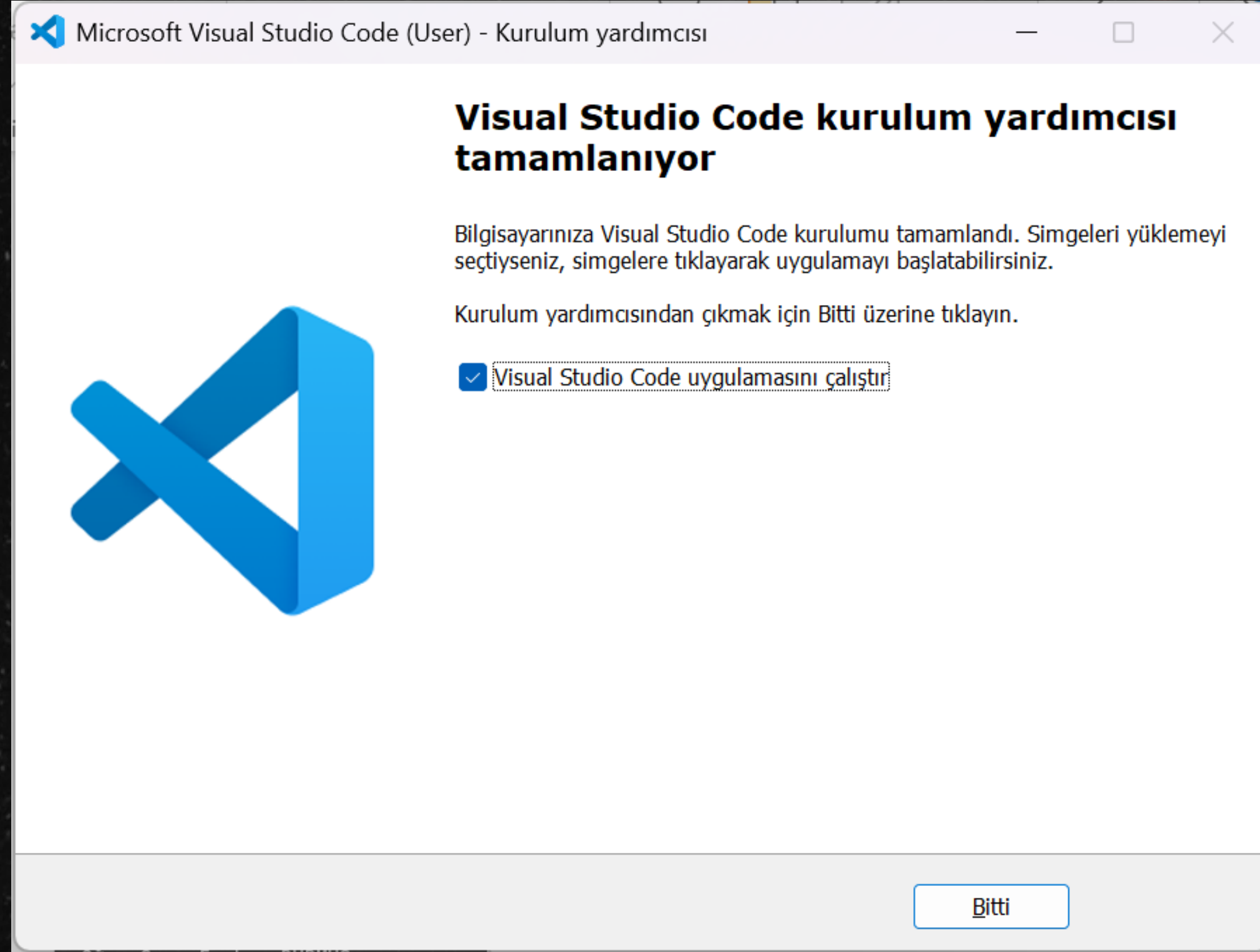
Getting started

indirme otomatik olarak başlamazsa buraya tıklayabilirsiniz

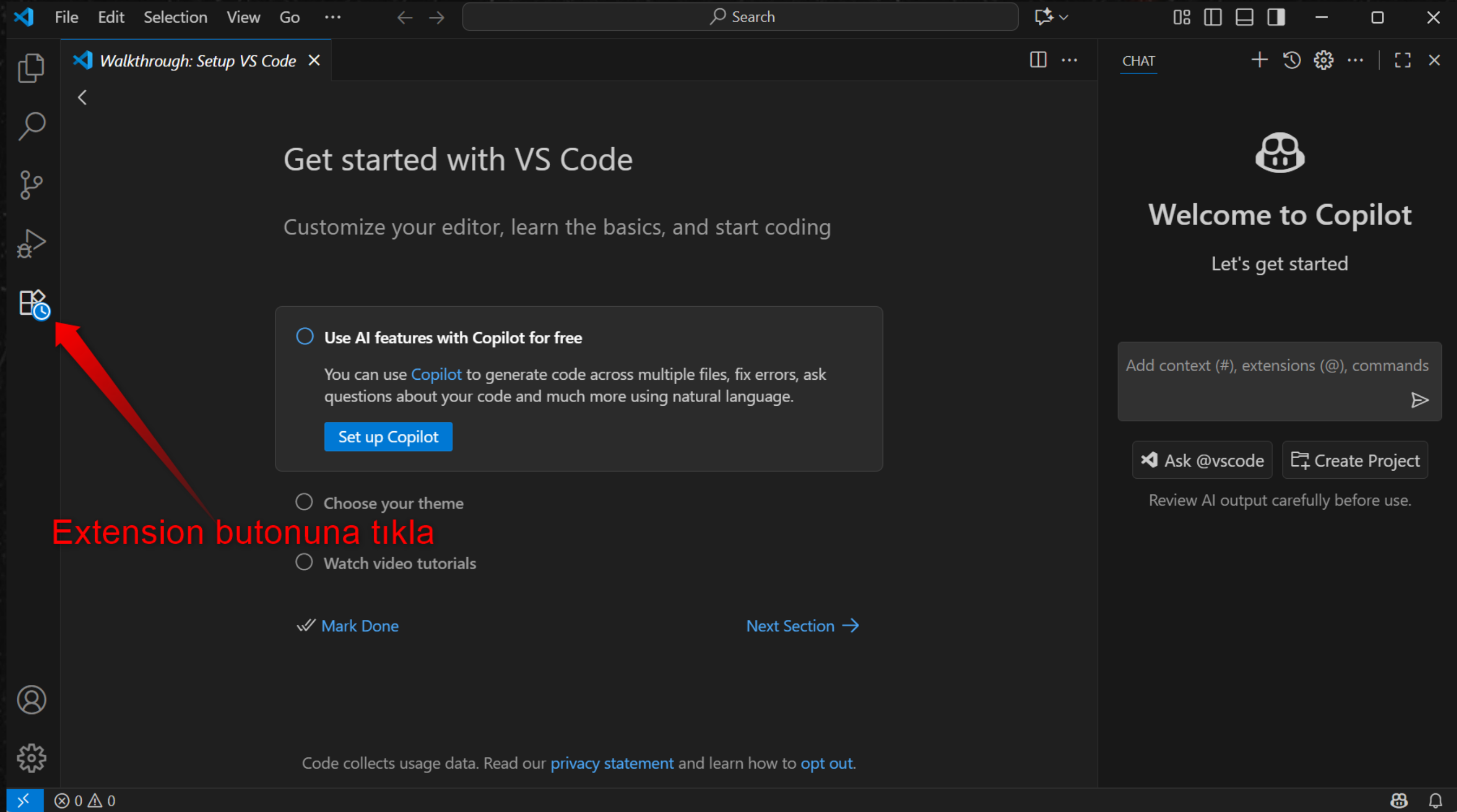
{05} IDE Kurulumu



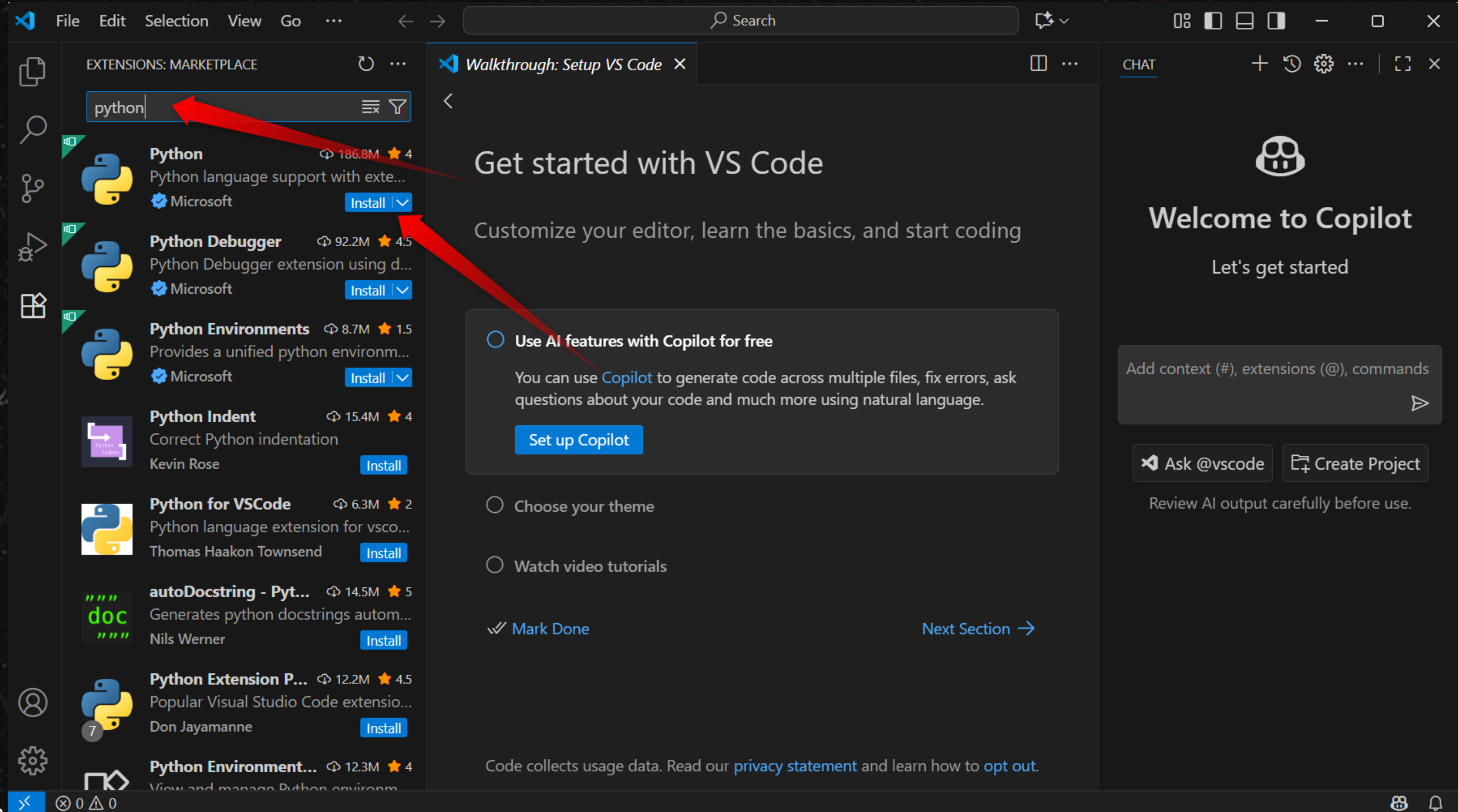
{05} IDE Kurulumu



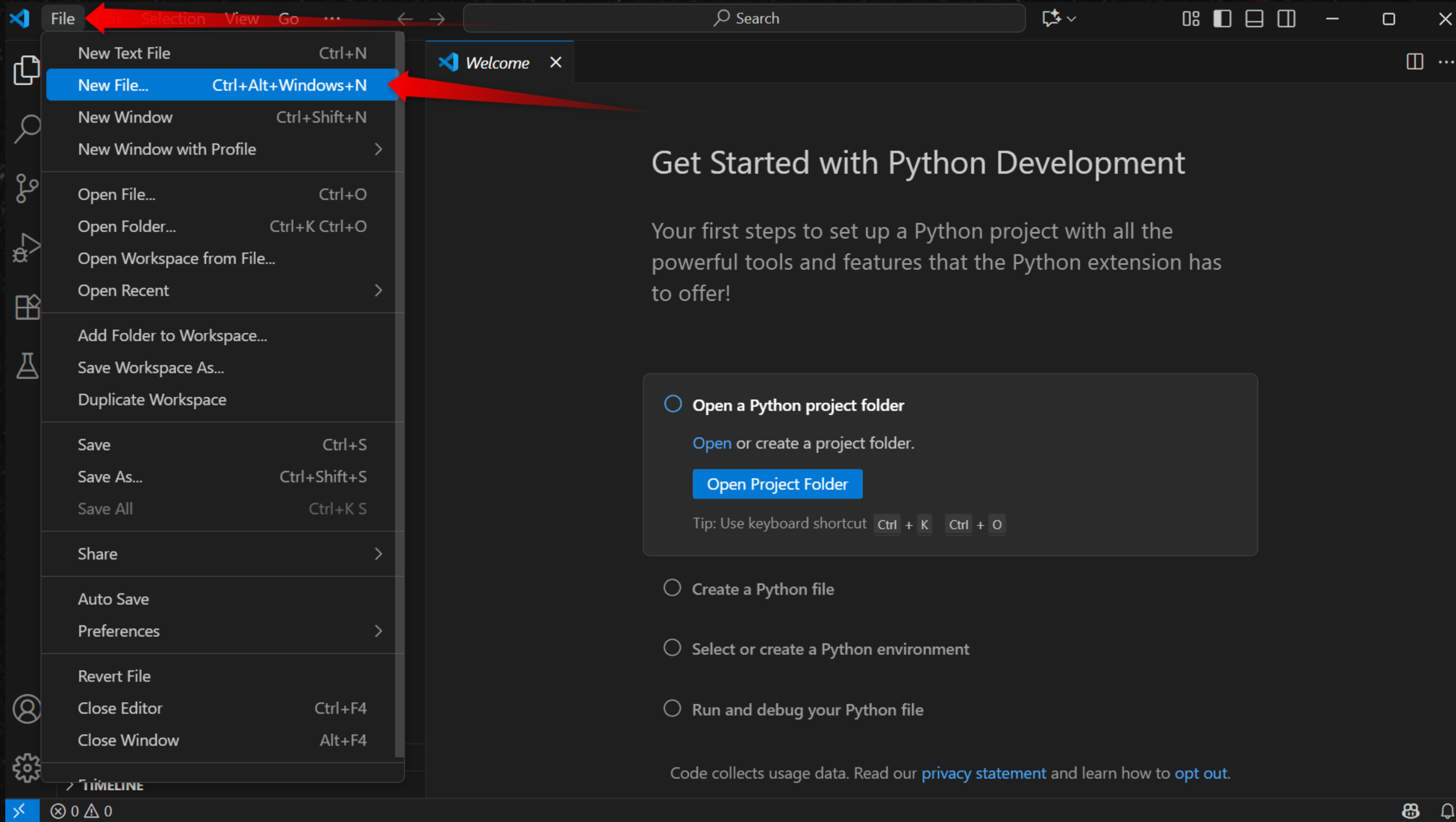
{05} IDE Kurulumu



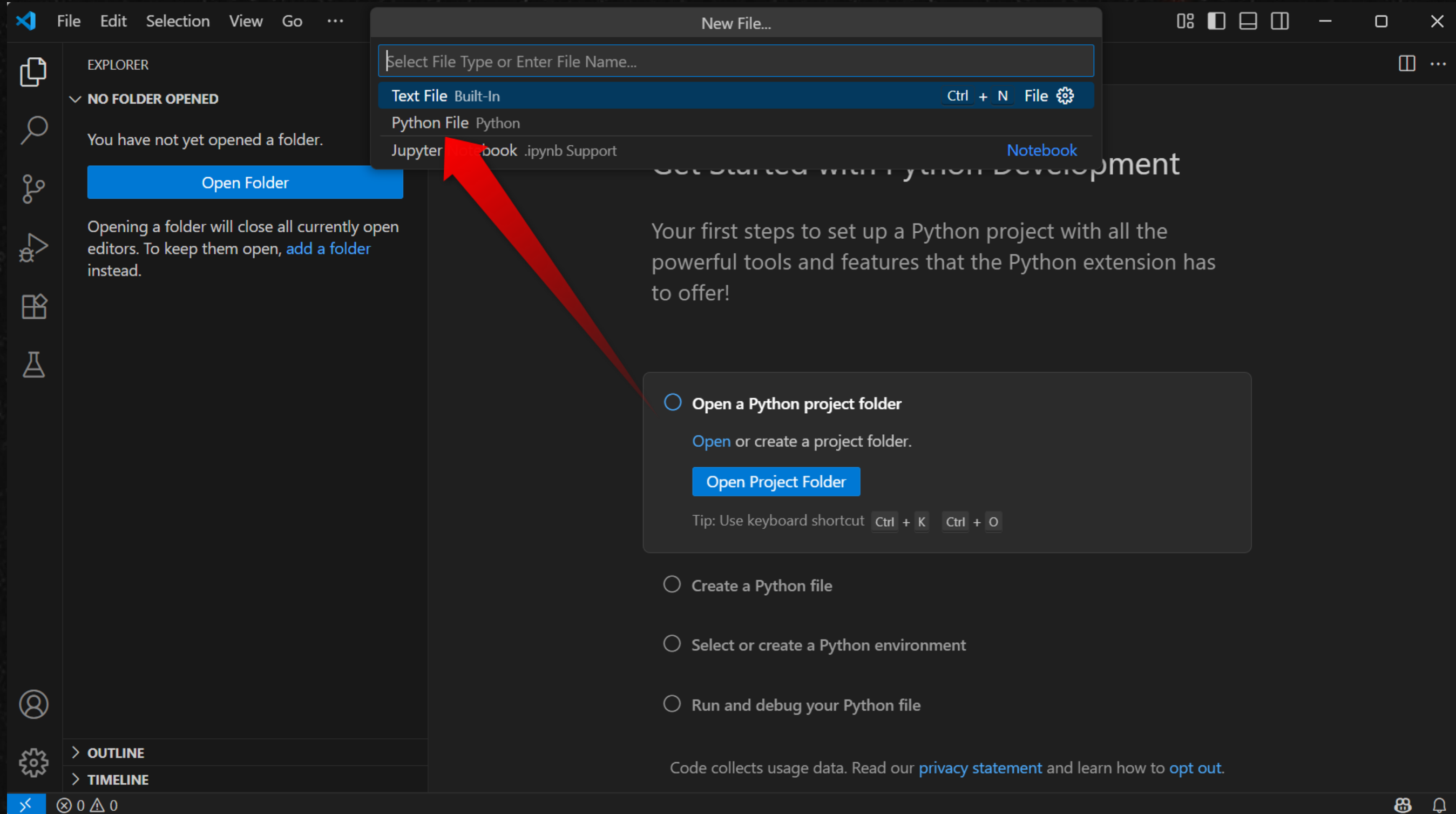
{05} IDE Kurulumu



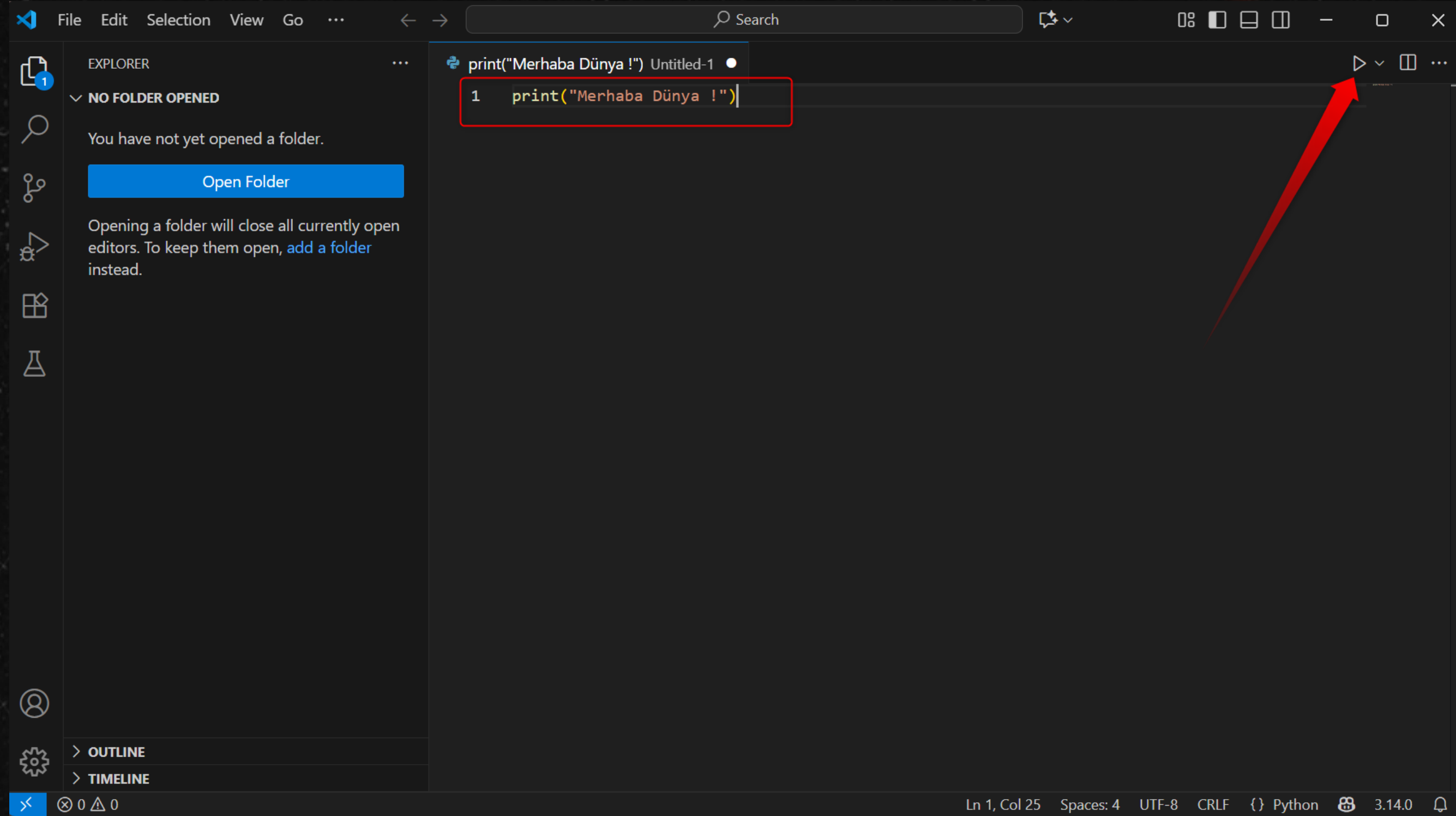
{05} IDE Kurulumu



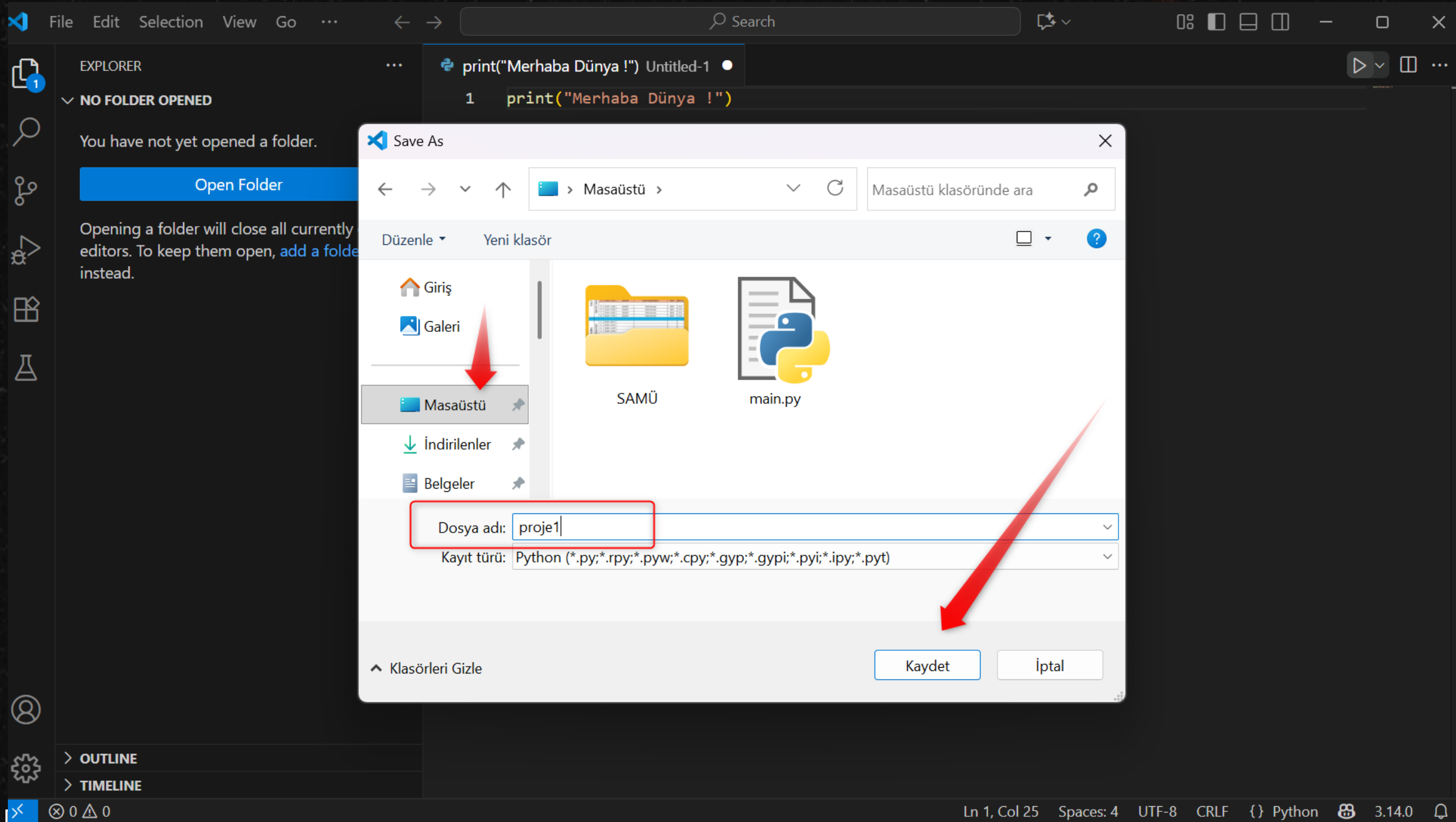
{05} IDE Kurulumu



{05} IDE Kurulumu



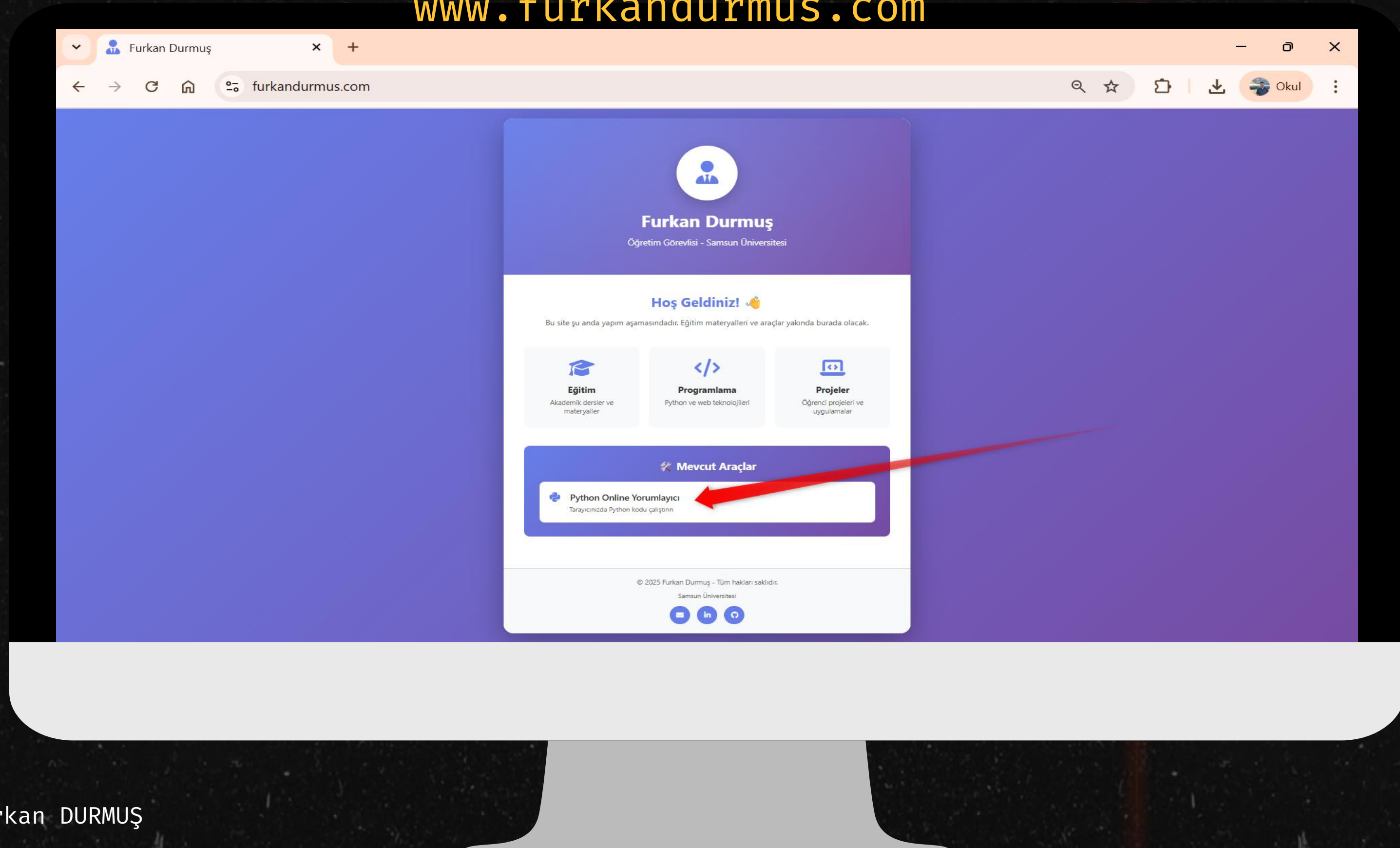
{05} IDE Kurulumu



{05} IDE Kurulumu

```
File Edit Selection View Go ... Search
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
proje1.py
C: > Users > SAMU-Furkan_Durmus > Desktop > proje1.py
1 print("Merhaba Dünya !")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + -
PS C:\Users\SAMU-Furkan_Durmus> & C:/Users/SAMU-Furkan_Durmus/AppData/Local/Programs/Python/Python314/python.exe c:/Users/SAMU-Furkan_Durmus/Desktop/proje1.py
Merhaba Dünya !
PS C:\Users\SAMU-Furkan_Durmus>
Spaces: 4 UTF-8 {} Python 3.14.0
```

www.furkandurmus.com



www.furkandurmus.com/python