

VS Code uygulamasında soldaki menüde üstten 5. sıradaki "Extensions" kısmında, "python" kelimesini aratıyoruz ve sonuçların üst kısmında "Microsoft" yazısı bulunan Pyhton u install ediyoruz. Bu extension python dosyalarını çalıştırmamızı ağlıyor.

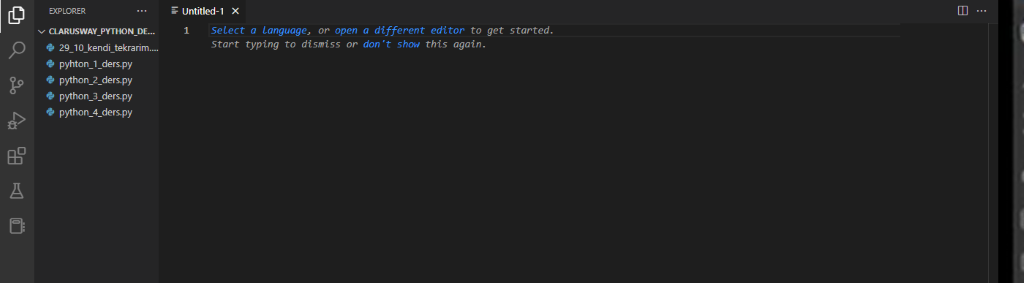
VS Code (Visual Studio Code) uygulamasında yeni bir klasör veya dosya oluştururken;

Dosyaya verdiğimiz isimlerde boşluk ve Türkçe karakter olmamalı. Dosya isminin yanına .py uzantısını eklersek VS Code bu dosyanın Python dosyası olduğunu anlıyor.

Üstteki menüden "Terminal" menüsüne tıkladığımızda açılan menüde "New Terminal" dediğimizde alt kısımda terminali açar. Bilgisayarda bulunan terminallerden birini açacaktır.



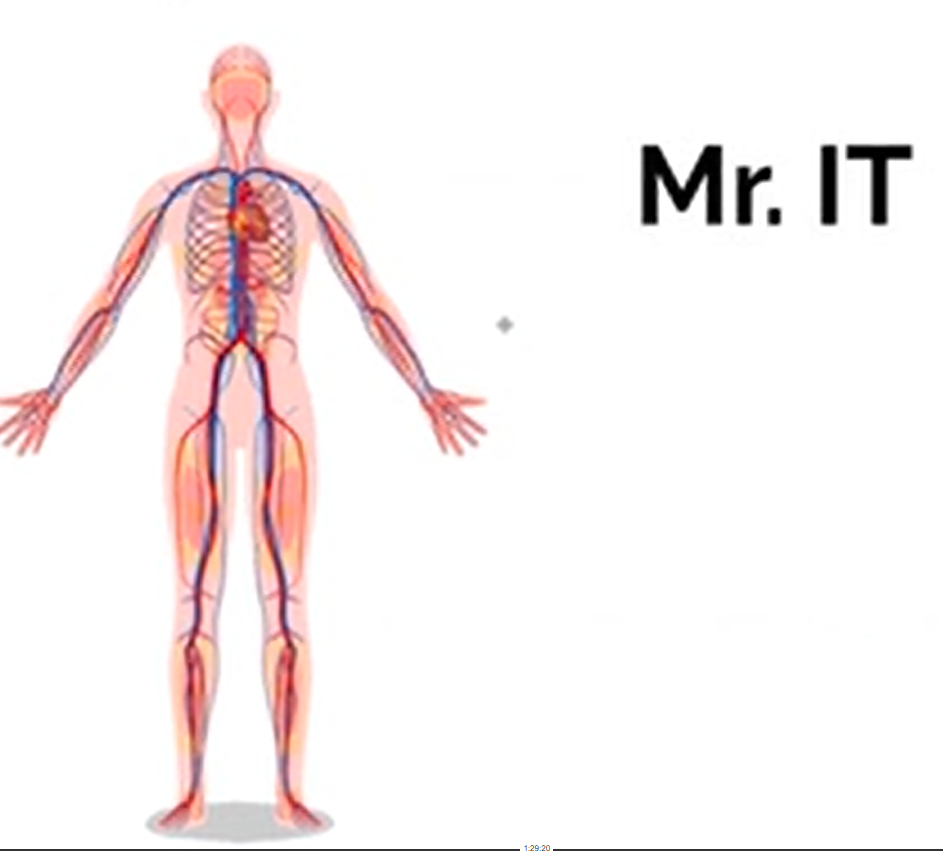
Terminalin sağ kısımnda "+" işaretinin yanındaki oka bastığımızda oradan terminal seçebiliriz. Herhangi bir terminali buradan default terminal yapabiliriz. Hocanın kullandığı terminal "Git Bash" terminali. Siz de onu kurduktan sonra bol bol onu kullanacaksınız dedi.



Yukarıdaki boş alana çift tıkladığımızda yeni bir dosya açılır. "Select language" e basarsak oradan istediğimiz programlama dilini de seçebiliriz.

Settings den "auto save" i aktif hale getirebiliriz. Ancak kod yazdıktan sonra kendimiz kayıt etmeyi alışkanlık haline getirmek daha iyi olur.

Python Basics Session-1

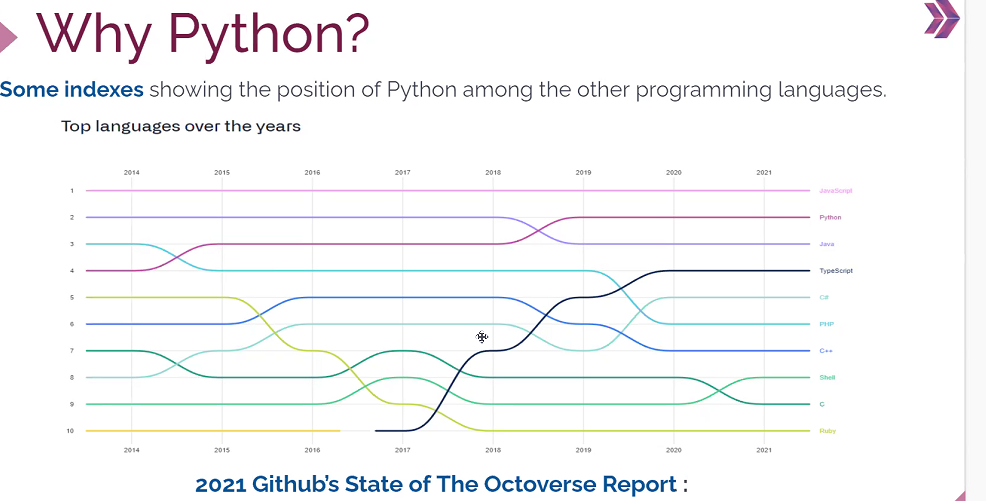


Pyhton Mr. IT nin damarlarında akan kandır.

Biz AWS ve Devops çu olarak AWS resource larını yönetirken Pyhton u kullanacağız.

Bir full stack çi kadar kod yazmayacağız ancak bir pthon script i yazacak duruma geleceğiz.

Hiç pre-class a bakmadan gelmeyin.



Pyhton ilk kez 1991 yılında ortaya çıkarılıyor ve dilin kullanımı kolay olduğu için yayılıyor. Halen en popüler dil ve en çok kullanılan ikinci dil. Günümüzde yapay zeka vb bir çok yerde kullanılıyor:

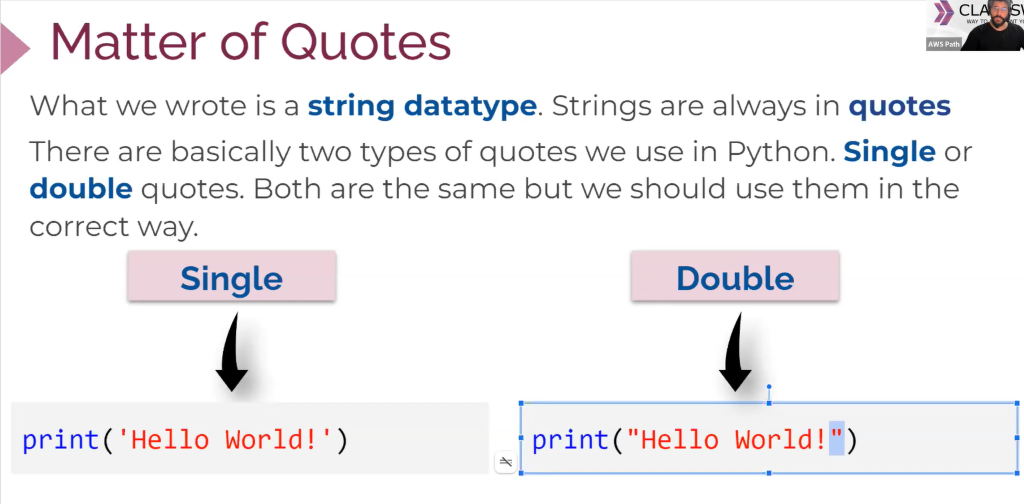




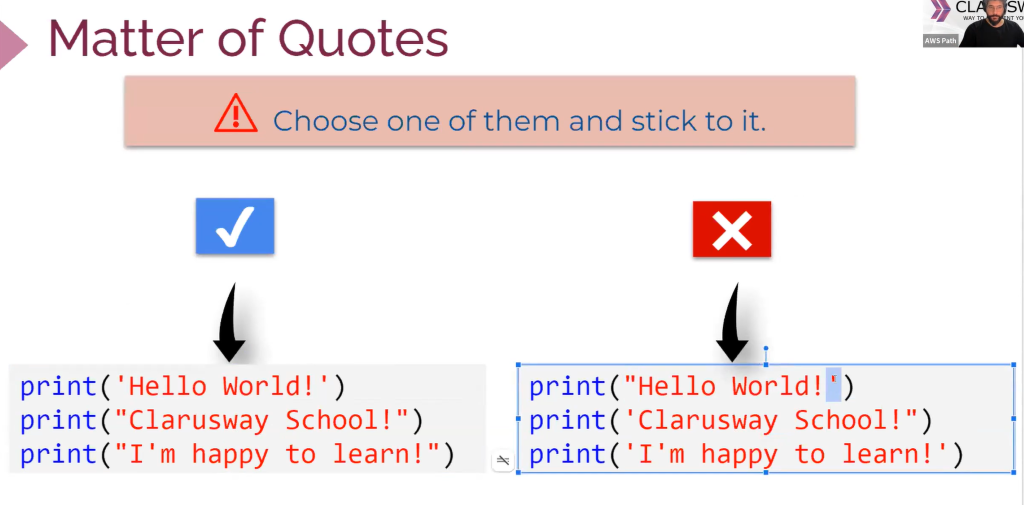
"Microsoft Developers on Youtube" videolarını takip etmemiz gerekiyor.

İlk yazdığımız kod "hello world" Bu internet bulunduğunda verilen ilk mesajdır. Bu nedenle bütün dilleri öğrenirken yazılan ilk kod budur.

Quotes



Tırnak içinde yazılan kodlar "String" datatype tır. Tek veya çift tırnak kullanabiliriz. İki tek tırnak ile çift tırnak farklıdır! Ancak dikkat edilmesi gereken nokta başlarken birisini seçip bitirirken de ona bağlı kalmalıyız:



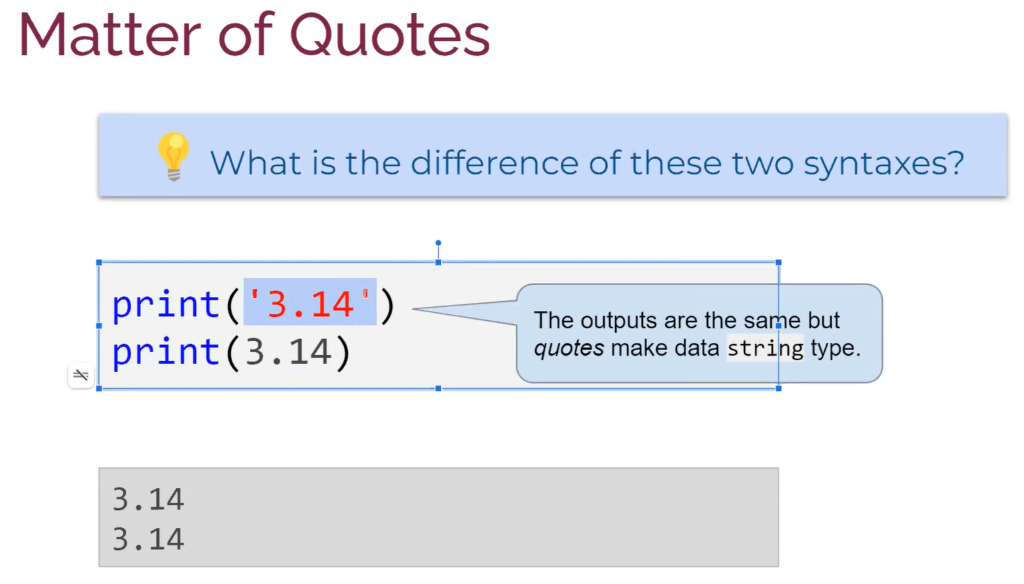
Son örnekte de görüldüğü gibi yazdığımız text içinde kesme işareti varsa bu metini çift tırnak ile yazmamız gerekir.

Üçlü tek tırnak ve üçlü çift tırnak kullanımlarının her ikisi de mümkün:

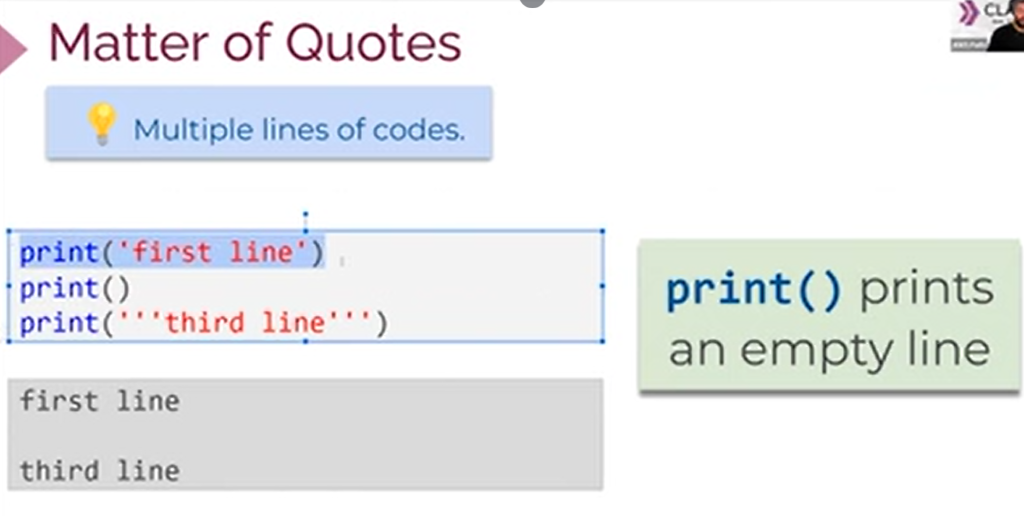
print('''Lorem ipsum dolor sit amet, fusce ut quisque neque donec, massa metus amet, luctus inceptos.''')

print("""Praesent a, mi mattis velit metus, accumsan adipiscing ipsum sit justo penatibus, amet mauris non tempus justo.""")

Üç tırnak kullanımı da tabiri caizse garantici kullanımdır. Ancak biz garantici olmayacağız.



Üstteki iki syntax aynı görüntüde çıktı verse de ikisinin type ı farklıdır. Tırnak içinde olan syntax lar her zaman string tir.

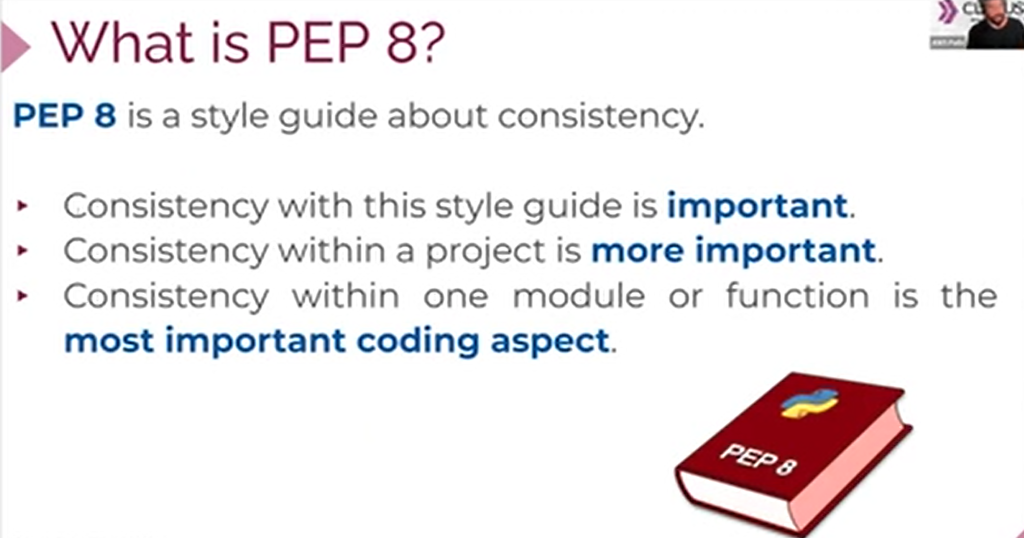


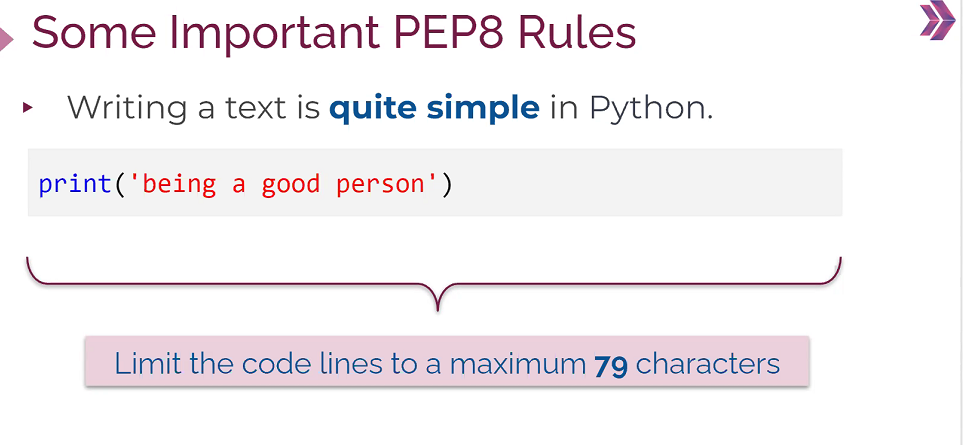
Print() komutu içi boş parantezin de çıktısını boşluk olarak verir.

PEP8 Rules (kuralları) Python'un ahlak kuralları gibidir. Bu kurallara uyulmazsa da kodlar çalışır.

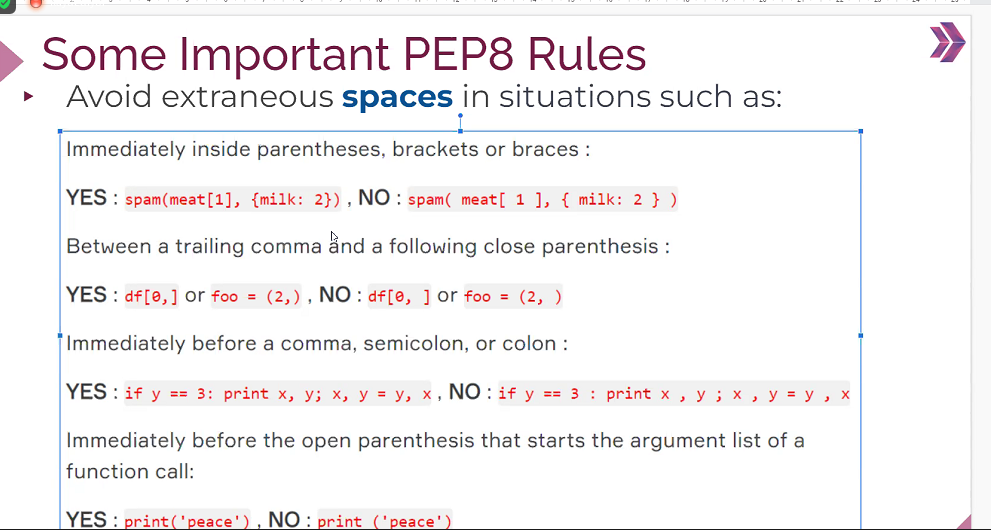
Ancak Python da yazılan kodların standart ve tutarlı hale gelmesi için, bizim veya başkaları tarafından daha rahat okunması açısından oluşturulmuş kurallardır.

PEP : Pyhton Enhancement Proposal





Bir code line maksimum 79 character olmalı. Görüntüsü daha güzel ve okunması daha kolay.

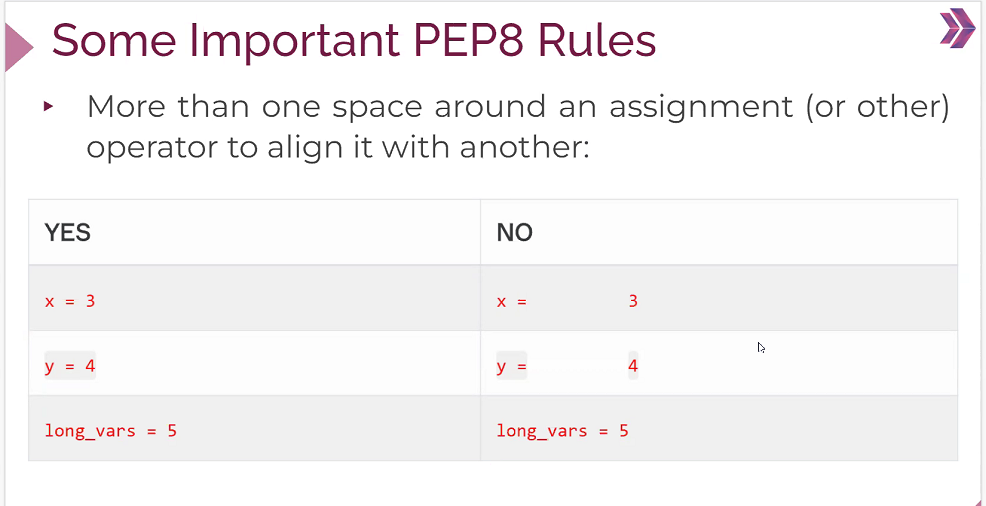


Parantez, köşeli parantez ve süslü parantez içlerinin bitişiğinde boşluk kullanmıyoruz.

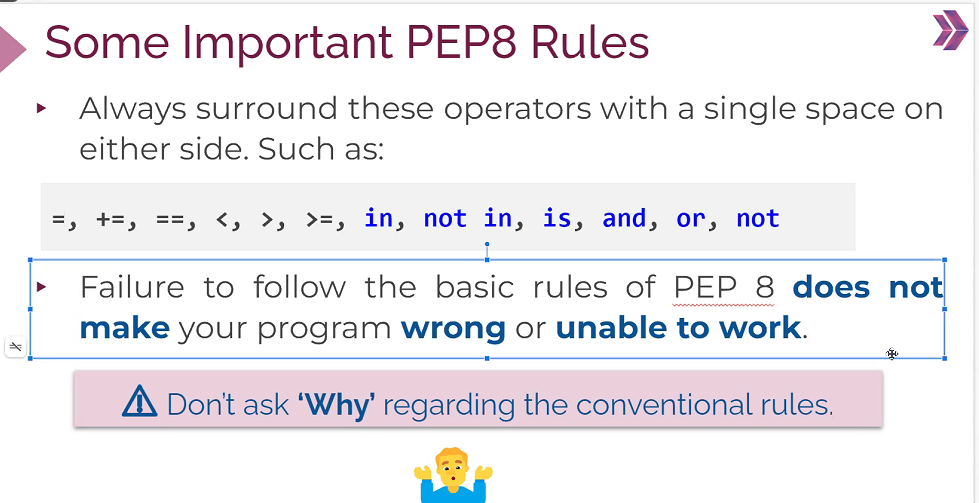
Virgülden sonra boşluk kullanırız ancak virgülden sonra parantezler kapanıyorsa boşluk kullanmıyoruz.

, ; : önce boşluk kullanmıyoruz.

Bir function call ile parantez arasında boşluk kullanmıyoruz.

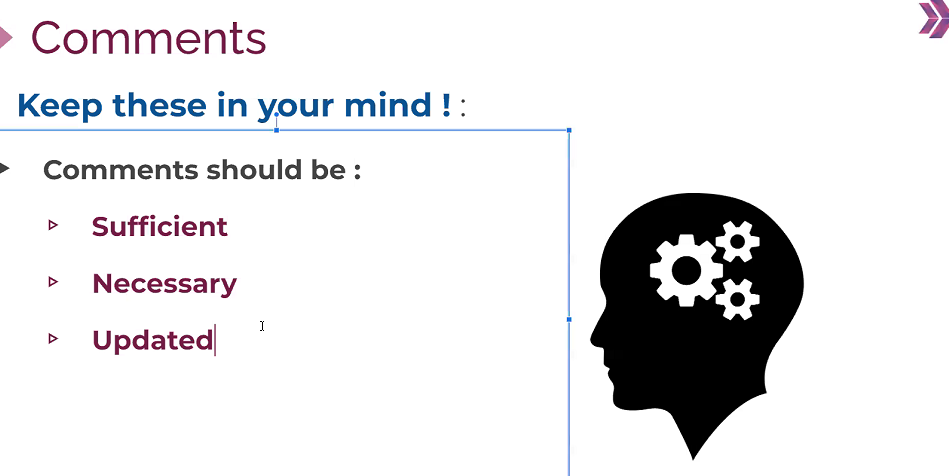


= işaretinden önce ve sonra boşluk bırakacağız. Satırlar aynı hizada olsun diye daha fazla boşluk bırakmayacağız.



Yukarıdaki anlamlı işaretlerin her iki tarafında birer boşluk bırakılır.

Bu kurallar için "neden" diye sormayın.



Yazdığımız kodları ileride ne geriye dönüp baktığımızda neden yazdığımızı hatırlayamayız. Başkaları da o kodu neden yazdığımızı anlayamayabilir. Bu nedenle commentler kullanırız. Commentler Yeterli, gerekli ve Güncel olmalıdır.

Öte yandan VS Code üzerinde çalışmalar yaparken notlar almak için de comment kullanırız.

Uzun uzun yazılan kodlar çalışmazsa; hangisinin çalışmadığını anlamak için de ctrl+ö kullanılarak hepsini commentleyip sonra her 5 satırı sırasıyla çalıştırıp hangi kodun çalışmadığı anlaşılır.

Comment için: öncesinde iki boşluk sonra # ve bir boşluk ile comment yazılır.