

Välkommen!

Välkommen till Python for Hackers!

Jag tänker lära ut grunderna i Python och programmering för att du skall kunna göra egna hacking verktyg och kunna förstå hur programmerare tänker när de skriver program. Du kommer att lära dig både programmering i python och hur man kan använda det när man hacker.

Viktigt

Det som är viktigt med övningarna är att man förstår dom, inte hur snabbt det går.

Eftersom detta är första gången som jag skriver en programmeringskurs så vet jag inte hur fort det går att göra övningarna, ibland kan jag göra dom för svåra eller för enkla.

Om en uppgift känns för svår eller konstig, prova att hoppa över den och prata med mig om den när jag kommer hem.

Om du tycker att jag har fel sorts övningar så säg till så ändrar vi det så att det blir som du vill ha dom.

Programmering (och hacking) är svårt att lära sig för alla så det kommer att ta emot ibland. Prata med mig så kan jag säkert förklara eller så listar vi ut hur vi tar oss framåt tillsammans.

ChatGPT är bra att ha för att kolla sin kod och analysera om något blir fel och att skriva kod. De flesta programmerare använder ChatGPT idag (men den gör inte alltid rätt)

Kursdelar

Jag har tänkt att dela in kursen i tre delar (vi får se om det håller).

- Grunderna i python, variabler, funktioner, enkla appar.
- Websidor i python, skriva egna websidor i python.
- Hacka websidor med python, prova att skriva hackverktyg som fungerar på sidorna som vi har gjort.

Kursprogram

Jag tänkte att vi kör i Windows med Visual Studio Code som editor. Om du vill köra i Kali så testas vi det också.

Andra saker än Python, jag tänker gå igenom andra saker än python. Verktyg som man använder i programmering (och hacking) som Git, Postman och Fiddler.

Men vi tar en sak i taget, först börjar vi med att sätta upp programmeringsmiljön.

Steg 1 – Programmeringsmiljön

Installera Python

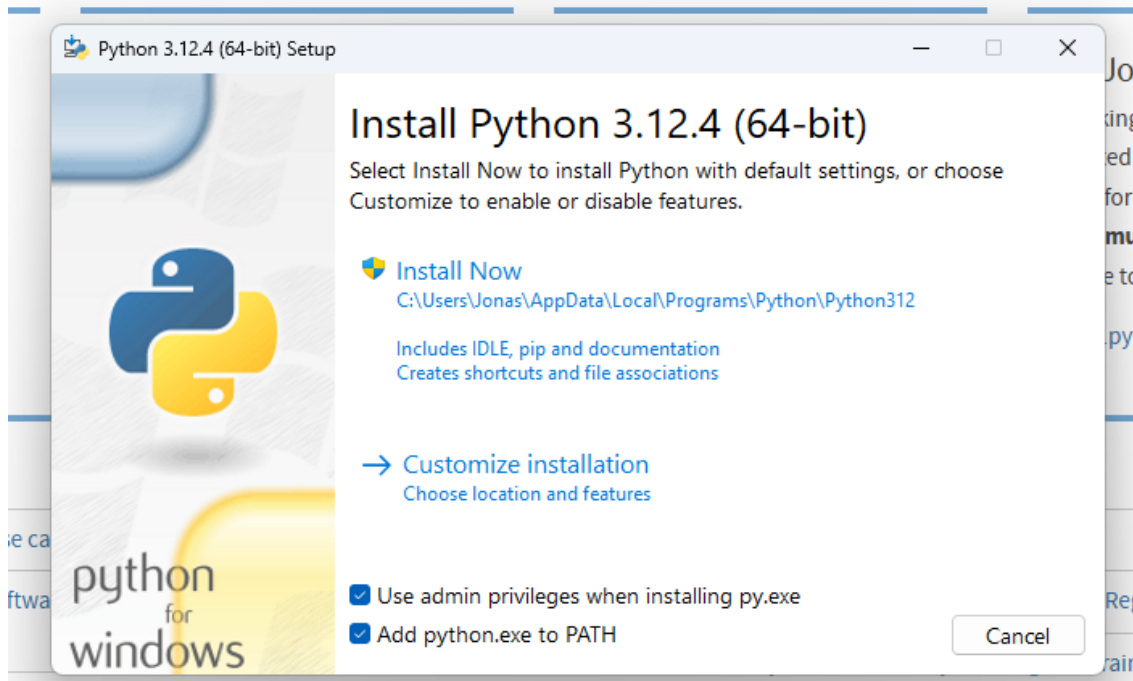
Ladda ner python på python.org (<https://www.python.org/>)

Ladda ner 3.12 versionen



Python for hackers

Kör filen och kryssa i bägge kryssen (Use admin och Add to Path)



Installera Visual Studio Code

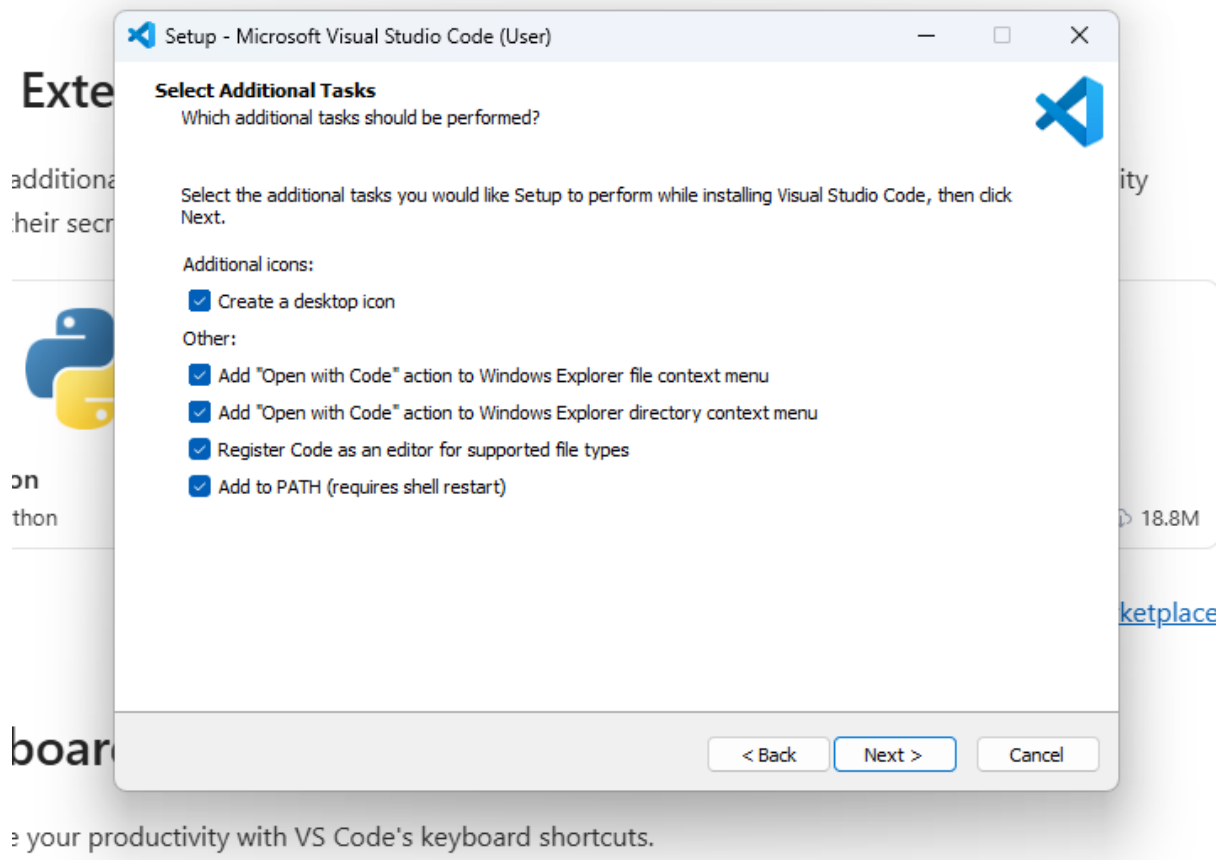
Installera Visual Studio Code från <https://code.visualstudio.com/Download>

Python for hackers

Kryssa i alla kryssen

[#1731](#) Learn how to write a VS Code extension.

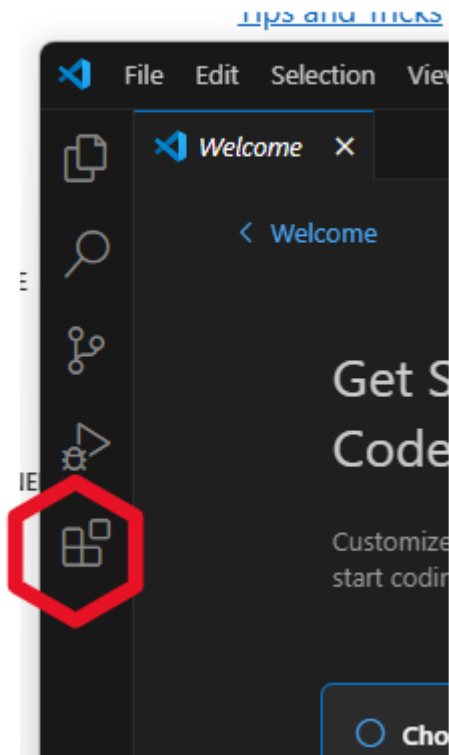
[Code?](#) - Read about the design philosophy and architecture of VS Code.



Starta Visual Studio Code om den inte redan är started

Python for hackers

Välj extensions



Installera Python (vänta till klar)

Installera Python debugger (vänta till klar)

Installera PyLance (vänta tills klar)

Så nu skall vi vara klara för att börja med övningarna

Övning: Hello world!

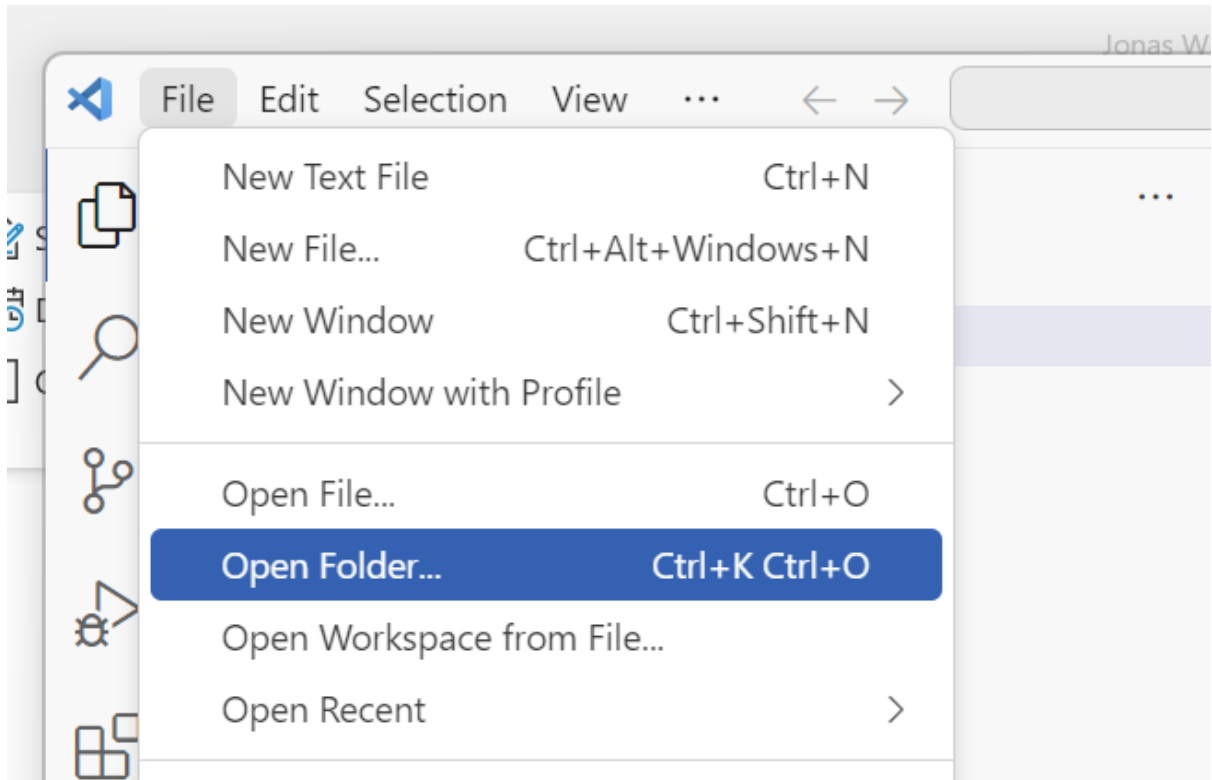
Denna kan du säker redan, men man brukar ju alltid börja med

Gör en mapp som heter PFH och en mapp under den som heter Code.

Python for hackers

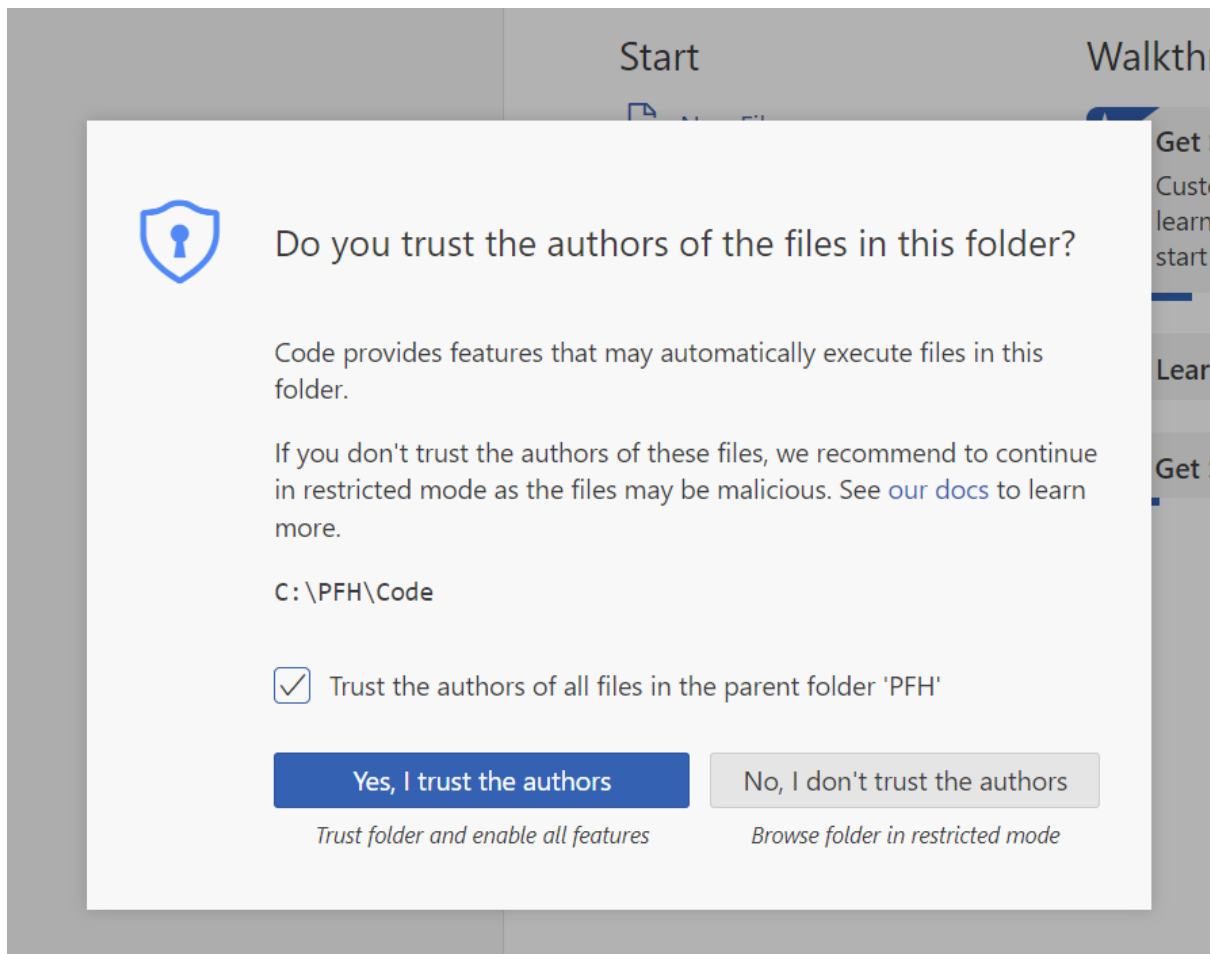
Starta Visual Studio Code

Öppna mappen PFH/Code med File|Open folder...

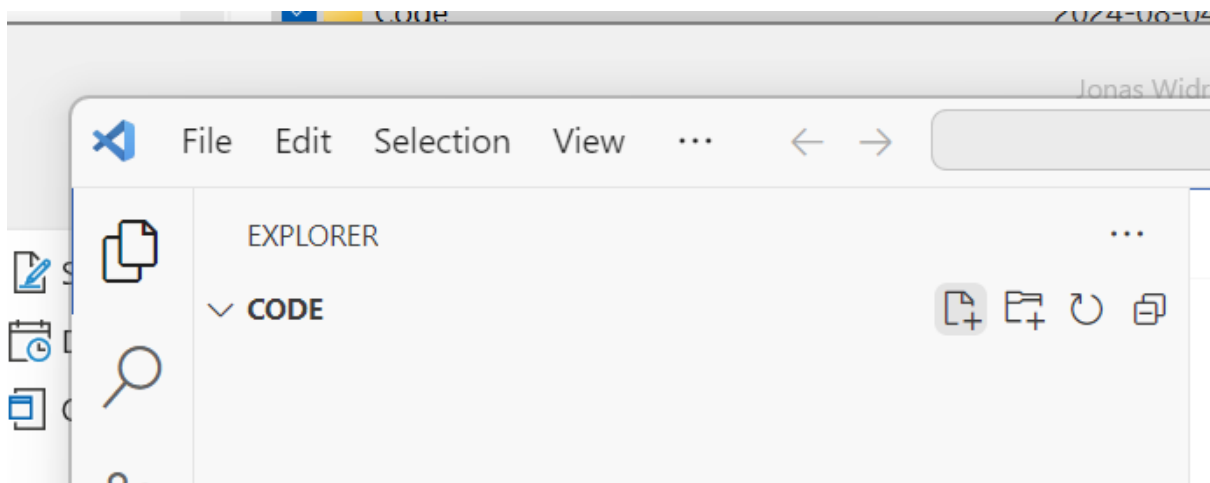


Python for hackers

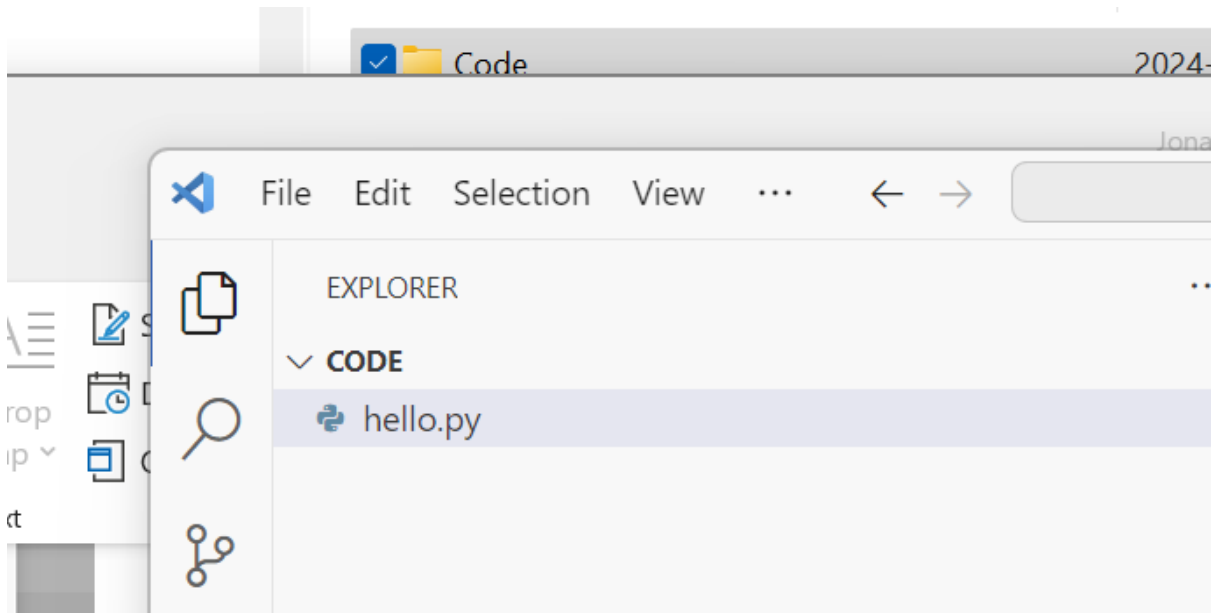
Kryssa i att du litar på mappen



Skapa en ny fil som heter hello.py



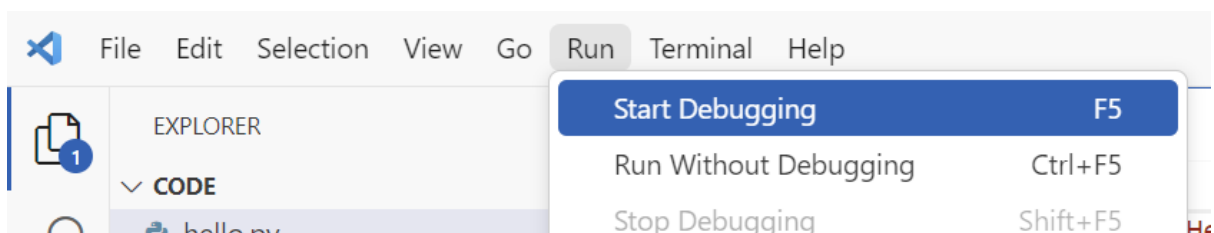
Python for hackers



I python finns en function som heter `print()`, vi använder den för att skriva ut en text.

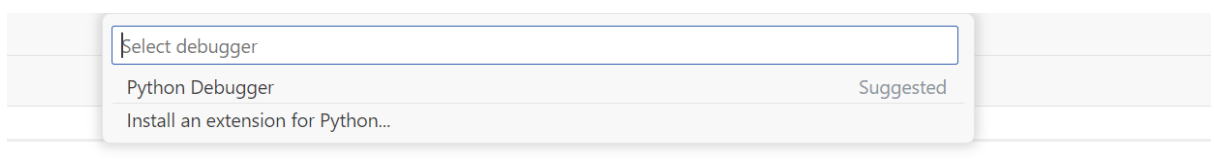
Skriv ut "Hello world"

Kör med F5 eller "Run|Start Debugging"



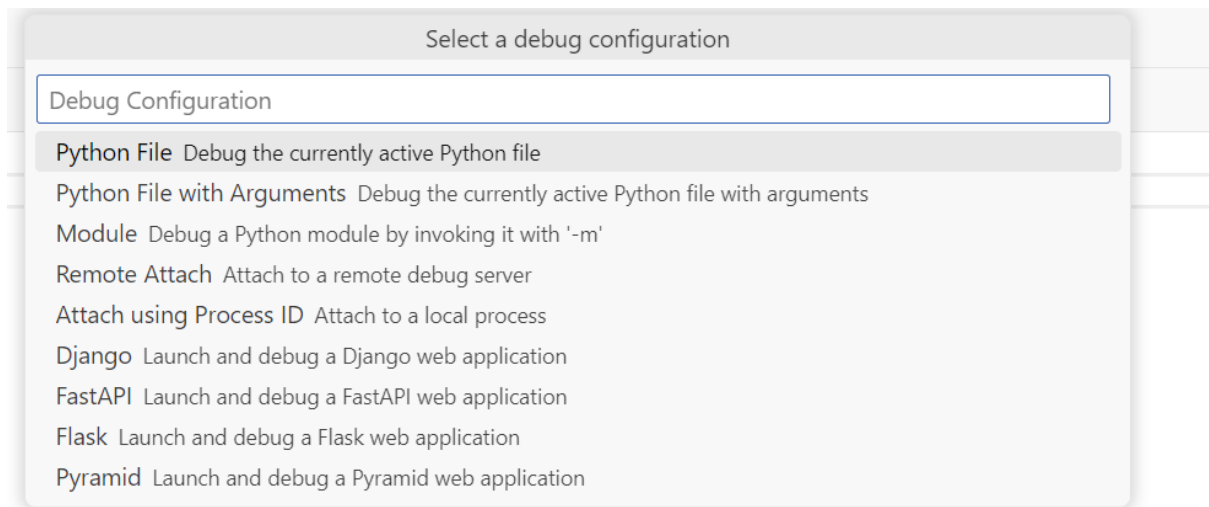
Visual Studio Code är ganska likt Visual Studio som du har använt tidigare.

Om du frågar om debugger så välj "Python debugger"



Python for hackers

Och sedan Python File på fråga 2.



Resultatet kommer nere i Terminal



Om du fick samma resultat skall allt vara rätt konfigurerat, grattis!

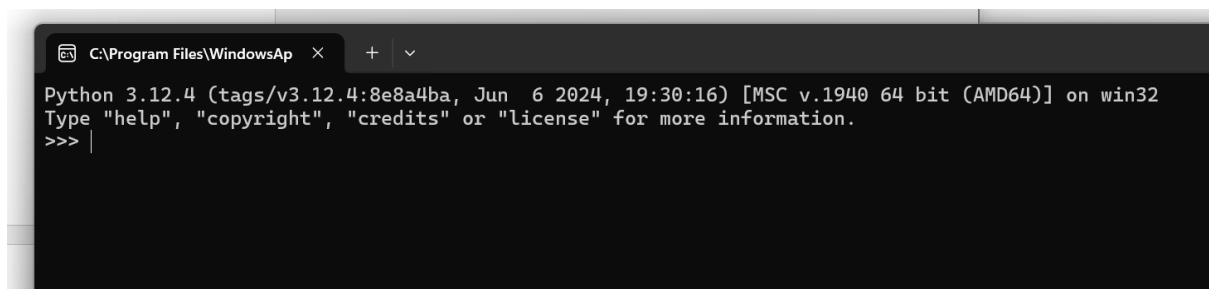
Övning Interaktivt läge

Python är interpreterande (tolkande om man översätter det lite), det betyder att Python program läses rad för rad till skillnad från C# som kompilerar.

Skillnaden är inte så stor men kan vara bra att känna till.

Man kan köra Python i ett interaktivt läge från kommandoprompten, det kan vara bra om man skall testa något lite snabbt.

Om du i kommandoprompten skriver "python" så kommer du in i det interaktiva läget (>>> betyder att du kan skriva)



```
C:\Program Files\WindowsAp x + v
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

Här kan man testa att göra variable och summara och testa en massa funktioner mm.

Vi testar att göra två variabler alpha och beta och plussa dom.

Python behöver inga variabel typer som C# utan listar ut det när man tilldelar värdet.

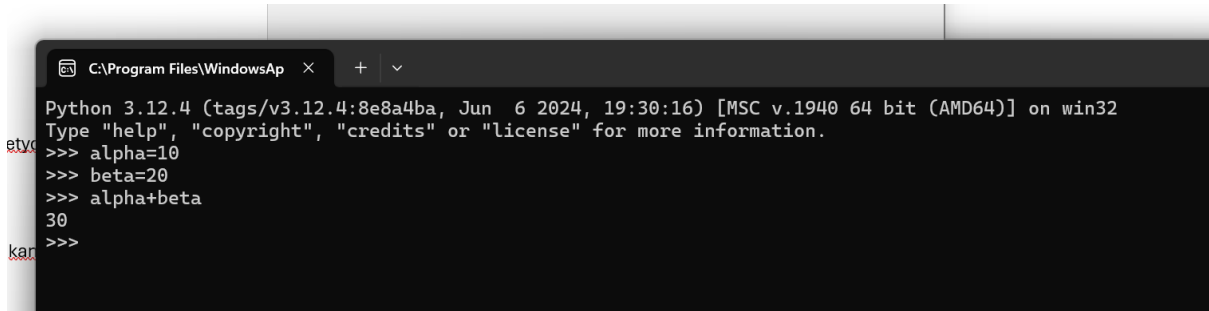
Vi skriver

```
alpha=10
```

```
beta=20
```

```
alpha+beta
```

Så här:



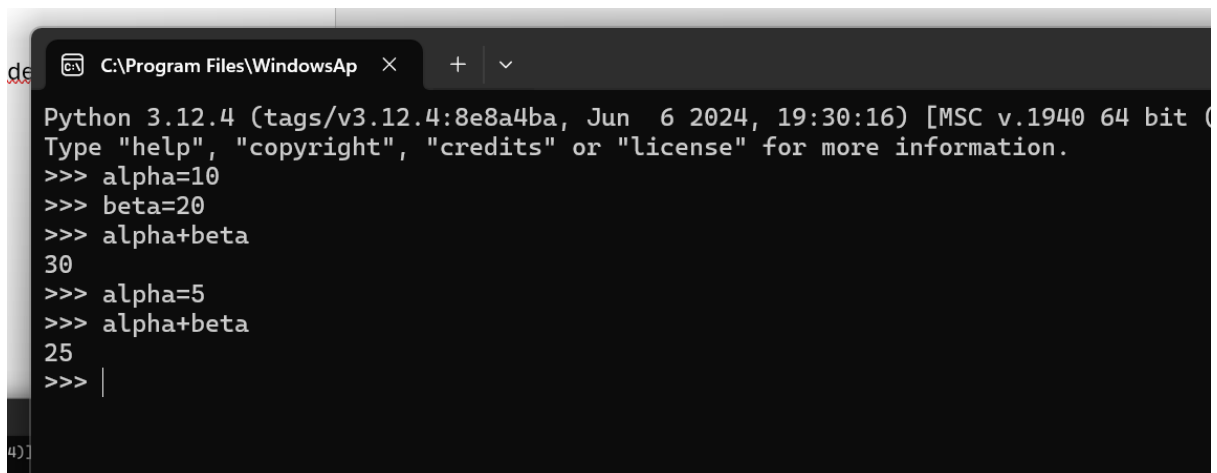
```
C:\Program Files\WindowsAp x + v
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> alpha=10
>>> beta=20
>>> alpha+beta
30
>>>
```

Nu kan du testa med

```
alpha=5
```

Python for hackers

alpha + beta



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license()" for more
>>> alpha=10
>>> beta=20
>>> alpha+beta
30
>>> alpha=5
>>> alpha+beta
25
>>> |
```

Så som du ser ändrade sig resultatet med ett annat alpha.

Du avslutar med quit() eller CTRL+Z.

Övning Strängar och input

Gör en ny fil som heter myname.py

Strängar i python ser ut så här:

```
s="En sträng"
```

eller

```
t='En annan sträng'
```

man lägger ihop två strängar med +

```
u="Första strängen " + "andra strängen"
```

Det finns en function som heter input()

(<https://docs.python.org/3/library/functions.html#input>) som skriver en prompt och lägger resultatet i en variabel tex:

```
veckodag = input("Vilken dag är det idag?")
```

Skriv ett program som frågar "Vad heter du?" till användaren, när användaren skrivit in sitt svar skall du skriva ut:

```
"Du heter Arne"
```

(om användaren skrev in Arne)

Glöm inte att klicka i terminal fönstret innan du skriver namnet.

Övning: Lägga ihop två tal

Vi skall göra ett program som lägger ihop två tal och skriver ut resultatet.

Gör en ny fil som heter mynumbers.py och i den skriver vi:

```
talEtt=input("Nummer1:")
```

```
talTvå=input("Nummer2:")
```

```
print(talEtt+talTvå)
```

Om du matar in 2 och 5, vad tror du resultatet blir?

Kör programmet och testa.

Varför blir det 25 och inte 7?

Jo, för att input ger en sträng så vi har slagit ihop två strängar, men vi vill slå ihop två tal.

För att göra om strängar (och andra tal) till tal så finns två funktioner `int()`

(<https://docs.python.org/3/library/functions.html#int>) och `floatP()`

(<https://docs.python.org/3/library/functions.html#float>)

Prova med att lägga `int` funktionen runt `input`-raderna så här:

```
talEtt=int(input("Nummer1:"))
```

```
talTvå=int(input("Nummer2:"))
```

```
print(talEtt+talTvå)
```

Python for hackers

Kör ingen och testa. Vad blev resultatet, det borde vara 7.

Övning: if-satser

If satser är viktiga i alla program, gör en fil som heter myif.py.

Vi skall testa med att göra ett program där man matar in temperaturen.

Om det är mindre än 10 grader skall den skriva ut kallt

När programmet är klart skall den skriva tre punkter.

```
temperatur=int(input("Hur många grader är det?"))
```

```
if temperatur<=10:
```

```
    print("Kallt")
```

```
print("...")
```

Som du ser är `print("Kallt")` inskjutet, det kallas ett block i Python och motsvarar `{ och }` i C#.

Prova med värdet 5 (som skriver ut Kallt) prova med 15 också och kolla att Kallt inte skrivs ut.

Vi lägger till den förklarande texten "(under 10)" efter Kallt om det är kallt.

```
temperatur=int(input("Hur många grader är det?"))
```

```
if temperatur<=10:
```

```
    print("Kallt")
```

```
    print("(under 10)")
```

```
print("...")
```

Eftersom den nya `print("(under 10)")` är inskjuten (ingår i blocket) körs den bara när villkoret är sant.

Python for hackers

Nu skall vi lägga till ett vilkor om det är varmare än 25 grader skall vi skriva två rader, "Varmt" och "(över 25)"

```
temperatur=int(input("Hur många grader är det?"))
```

```
if temperatur<=10:
```

```
    print("Kallt")
```

```
    print("(under 10)")
```

```
elif temperatur>=25:
```

```
    print("Varmt")
```

```
    print("(över 25)")
```

```
print("...")
```

Så där kör och prova.

elif är som else if i C#.

De två print-raderna efter elif radeden är ett block.

Nu skall vi lägga till texten "Normalt" och "(mellan 10 och 25)" för de andra temperaturerna.

```
temperatur=int(input("Hur många grader är det?"))
```

```
if temperatur<=10:
```

```
    print("Kallt")
```

```
    print("(under 10)")
```

```
elif temperatur>=25:
```


Python for hackers

```
print("Varmt")  
print("(över 25)")
```

else:

```
print("Normalt")  
print("(mellan 10 och 25)")
```

```
print("...")
```

Kör det och prova med lite olika värden.

Nu har vi gått igenom if-statsen

if *villkor*:

elif *villkor*:

else:

Lika med är som i C# ==.

Gör en ny fil som heter mynames.py,

Gör ett program som ber användaren att mata in ett namn.

Om namnet är Axel skriv Son, Melina skriv Dotter, Anna skriv Mamma och Jonas skriv Pappa. Om det inte är något av namnen skriv Okänd.

Övning: Kalkylator

Nu har du lärt dig tillräckligt många kommandon för att skriva en kalkylator på samma sätt som du gjorde i C#.

Man skall kunna mata in två tal, sedan skall man få frågan om man vill göra +, -, * eller /.

Sedan skall resultatet visas.