

Introduktion till programmering

DA354A 7.5HP HT20 LP2

Surfa till www.menti.com, använd koden:
17 98 59 7

TRANSFORMATION



Digitalisering avser ursprungligen och i tekniska sammanhang omvandling från analog till digital representation av information med nollor och ettor, exempelvis genom sampling av ljud och mätsignaler eller genom skanning av bilder.



Ett informationssamhälle





En väg framåt i utvecklingen

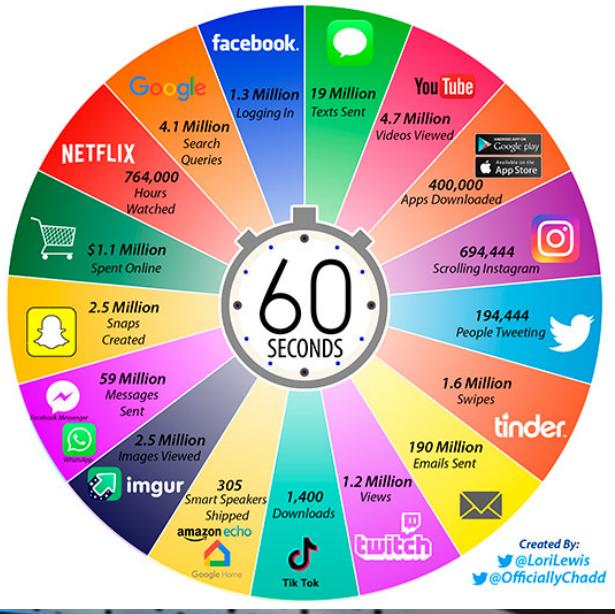
2020 This Is What Happens In An Internet Minute



Created By:
@LoriLewis
@OfficiallyChadd

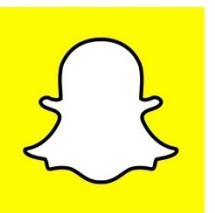
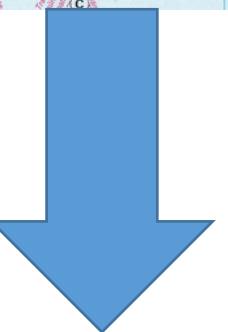
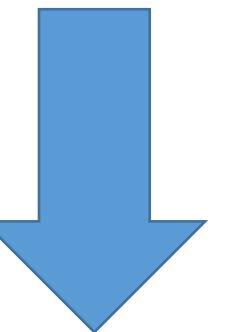
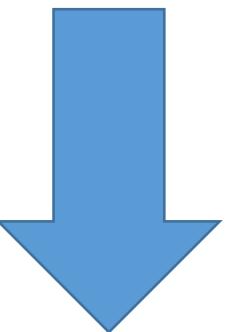


2020 This Is What Happens In An Internet Minute









Old School.



**Highly
Efficient.**

VS.





NETFLIX



NETFLIX ORIGINAL

OZARK

[Spela upp](#) [Min lista](#)**Se säsong 2 nu**

När det krisar måste familjen hålla ihop. Myskväll med familjen, det är nu att proppa väggarna fulla med knarkpengar.

ORIGINALINNEHÅLL FRÅN NETFLIX

Exempel: Skånetrafiken

- Här lägger jag specifikt fokus på biljetter som företeelse
 - Pappersbiljetten
 - Plastkortsbiljetten
 - Appbiljetten
 - Digitalt beroende verksamhet, värde kan svårlijgen skapas utan IT.

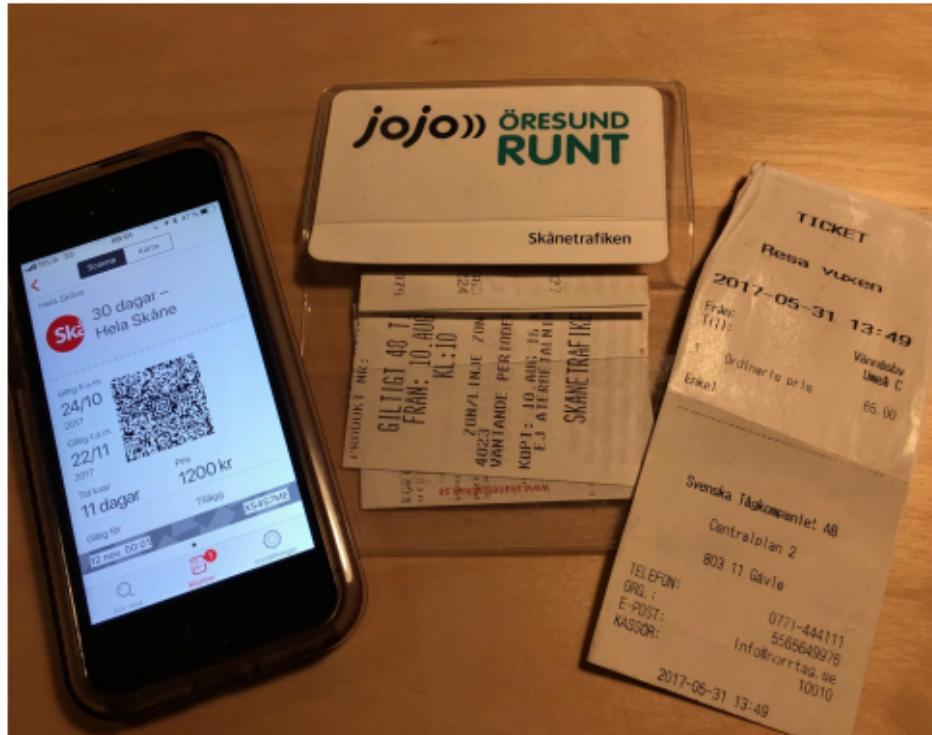
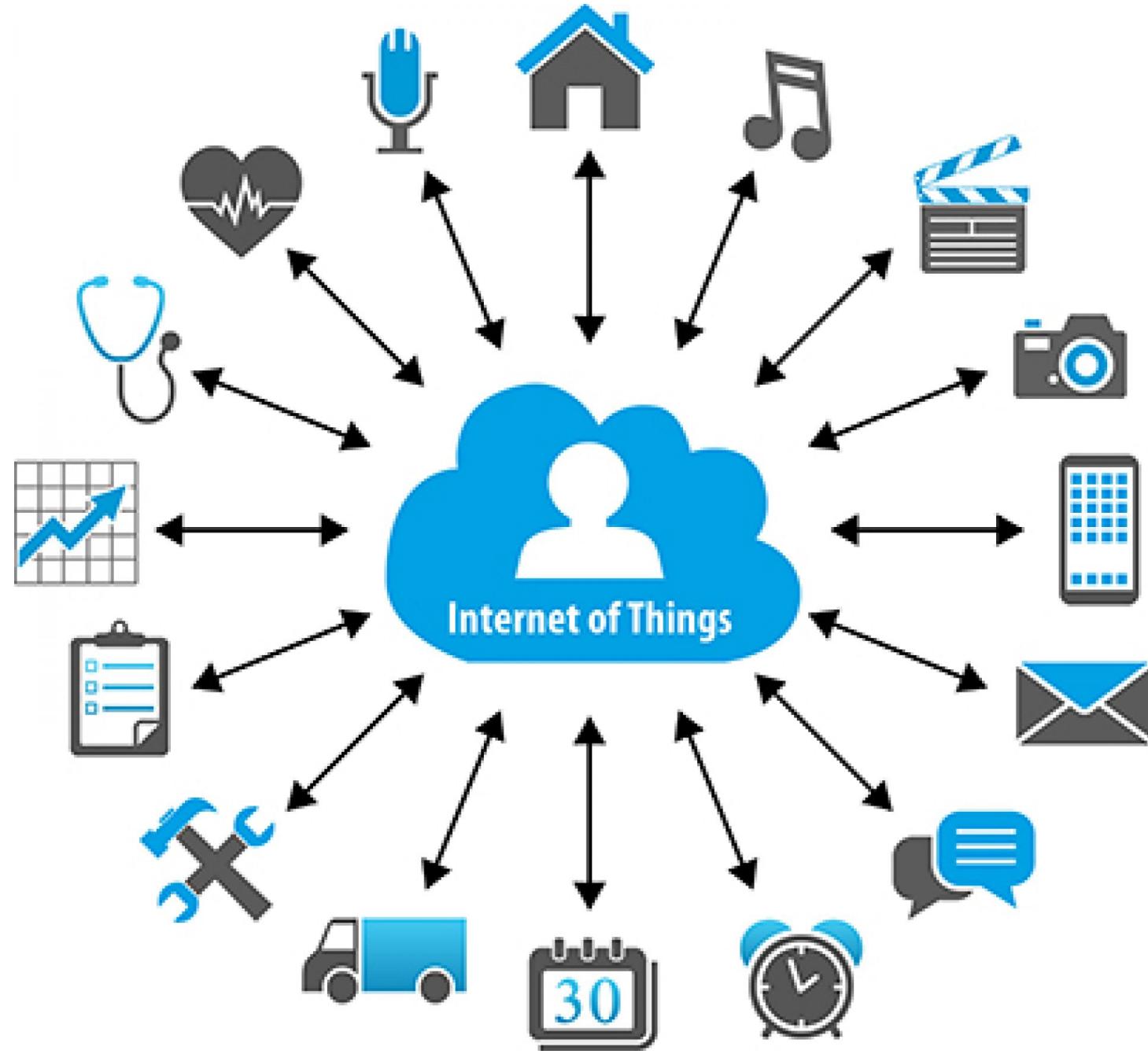


Foto: Carljohan Orre













HOW DOES IT WORK



SVENSKA DAGBLADET

[Start](#)[Näringsliv](#)[Kultur](#)[Ledare](#)[☰ Meny](#)[Debatt](#)

Programmering är framtidens språk

Det är hög tid att fler unga får bekanta sig med programmering tidigt i livet. Det här är inget nytt. Flera länder har redan insett det och det är dags att även Sverige gör det, skriver Christine Johnson, IT-Gymnasiet.

BRÄNNPUNKT | IT I SKOLAN

”

I dag finns språket kod i nästan all teknik vi äger och använder oss
av.

Christine Johnson

**Var med och
påverka!**

Uttryck dina åsikter
om reklam, media och
varumärken.

Anmäl dig här

För att Sverige ska stärka sin konkurrenskraft och utvecklas till den kunskapsnation vi måste bli så behöver vi alla lära oss ett nytt språk: programkod (kod). Likaså viktig som engelskan varit de senaste 75 åren, kommer kod att bli för oss om bara några år.

Debatt

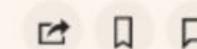
Det här är en argumenterande text med syfte att påverka.
Åsikterna som uttrycks är

Ekonomistudion

Brist på programmerare – saknas en miljon i EU

SvD Näringsliv

Publicerad 2018-04-23





DEBATT



Så skickar du in dina debattartiklar!

Mejla inläggen till debatt-redaktionen

MEST LÄST PÅ DEBATT I DAG



Sätt farliga elever i särskilda skolor

Fridolins köttnationalism hotar miljön och klimatet

Kod är makt och vi är framtidens analfabeter

Publicerad 22 okt 2014 kl 06.10

f Rekommendera

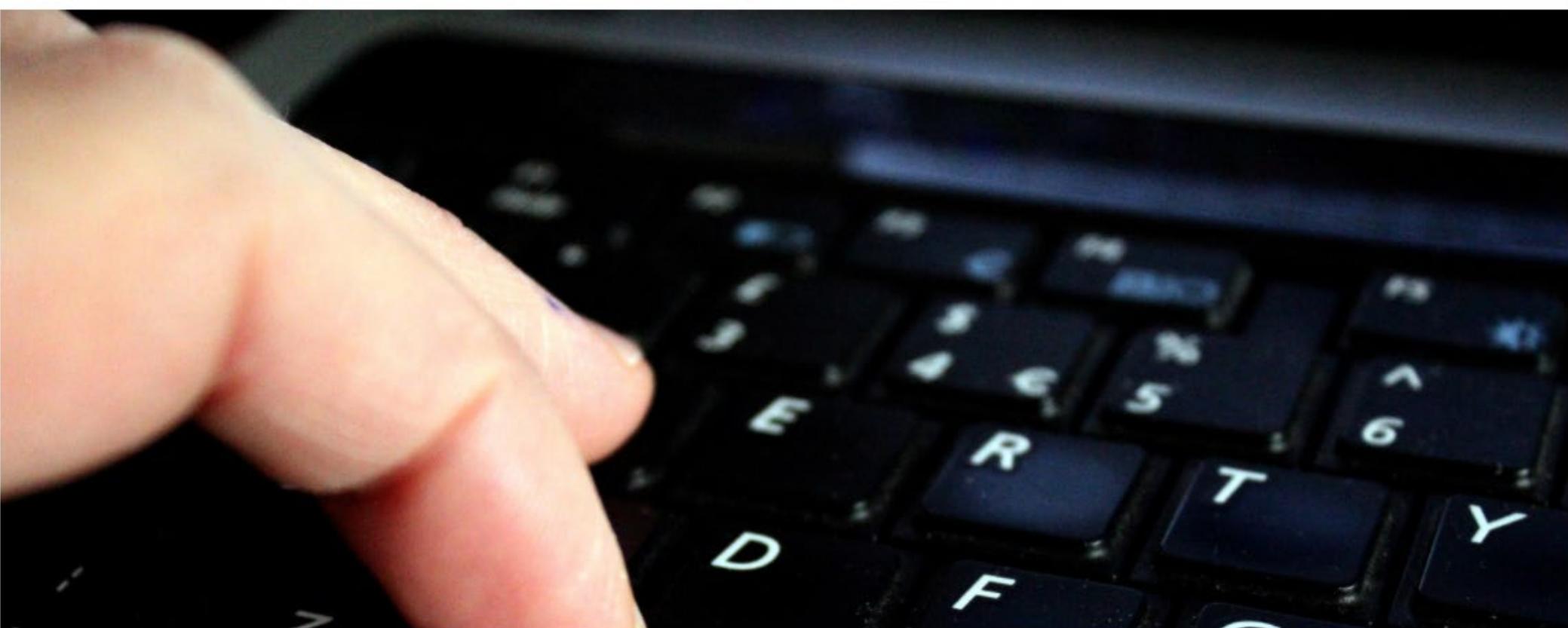
502

t Tweeta

g+ Dela

m Mejla

Den stereotypa 50-åriga universitetsutbildade mannen har fått konkurrens i



Programmerare

[Start](#) / [Yrken](#) / [Data & IT](#) / [Programmerare](#)

År 2020 kommer Europa att sakna 1.000.000 programmerare (enligt prognos från EU-kommissionen).

Programmerare kan arbeta med allt ifrån dataspel till avancerade styrsystem för tillverkning i industrin. Vanligen arbetar du utifrån riktlinjer eller instruktioner som systemutvecklaren dragit upp eller diskuterar själv förutsättningarna med beställarna av programmet. Du måste vara beredd att lära dig nya språk när det krävs.

Genomsnittlig lön* (Källa: SCB)

* Angiven lön kan variera kraftigt beroende på efterfrågan, utbildningsnivå, yrkesfarenhet etc.

42 800 kr

↑ Lätt att få jobb!





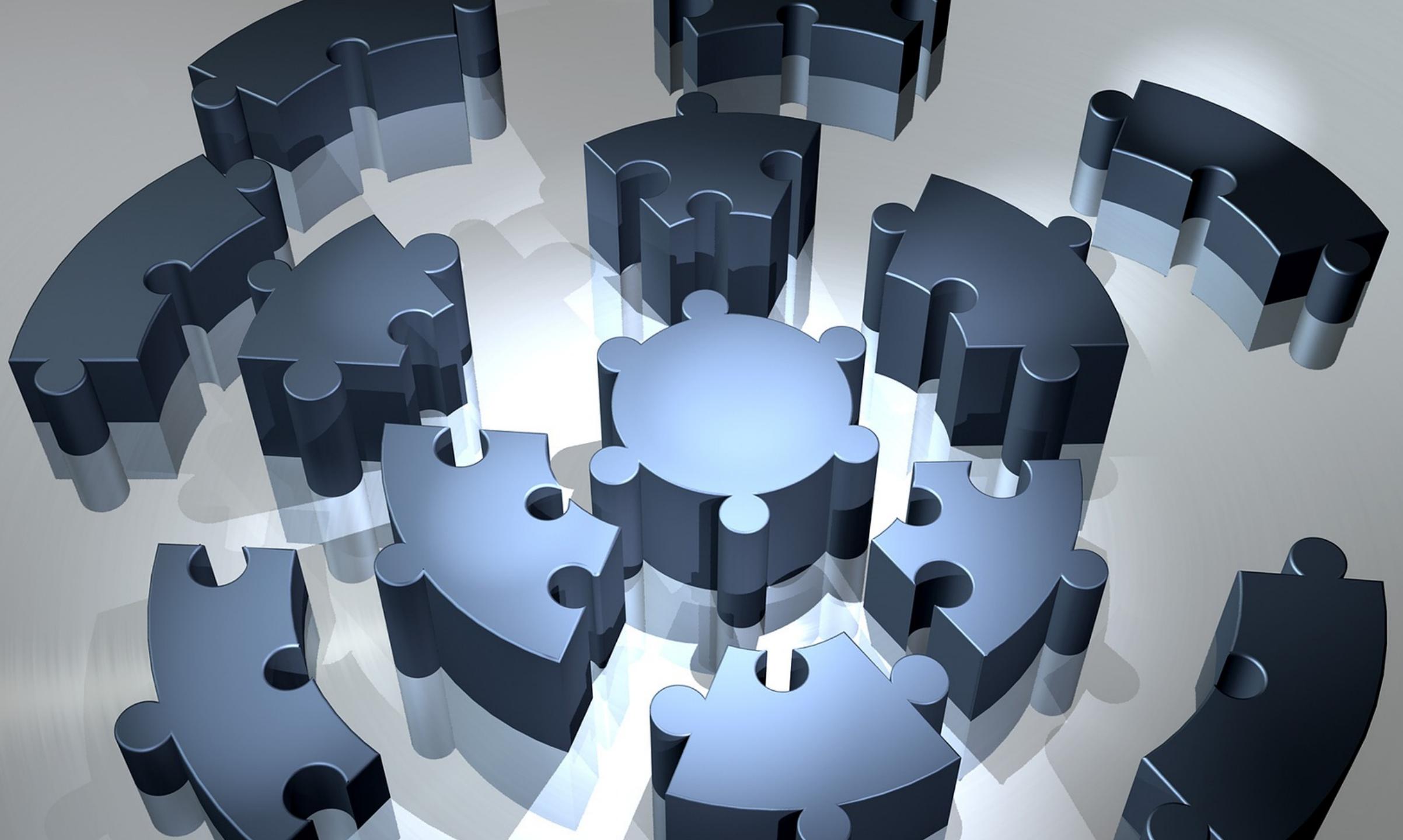
TALK
<CODE>
TO ME



TALK
<CODE>
TO ME



Vad går kurserna ut på?





A close-up photograph of a person's face, focusing on their eyes and nose. They are wearing dark-rimmed glasses. The person is looking directly at a computer screen whose content is visible through the lenses of the glasses. The screen displays a large amount of code, likely JavaScript or similar programming language, with various functions, variables, and syntax highlighted in different colors (e.g., blue for functions, red for strings). The background is dark, making the light from the screen stand out.

```
st a"), f=a.Event("hide.bs.tab  
faultPrevented()){var h=a(d);  
trigger({type:"shown.bs.tab",  
u > .active").removeClass("ac  
ria-expanded",!0),h?(b[0].off  
().find('[data-toggle="tab"]  
de")||!d.find("> .fade").l  
;var d=a.fn.tab;a.fn.tab=  
"show");a(document).on("br  
use strict";function b(b)  
=typeof b&&e[b]())}var  
",a.proxy(this.checkPo  
null,this.pinnedOffset  
tState=function(a,b,c,  
"bottom"==this.affixed  
!-=c&&e<=c?"top":null  
.RESET).addClass("af  
WithEventLoop=funct  
ent.height(),d=this  
peof e&&(e=d.top(t  
ent.css("top",""
```

Dagens agenda

- ~~Upprop~~
- Lärare & assistenter
- Kursens innehåll
- Kursplan
- Kursens planering
- Examinerande moment
- Kurslitteratur
- En historik programmeringsresa – Med Johan Holmberg



Lärare & assistenter

Anton Tibblin



Lärare

Amanda Tell



Assistent

Kursens syfte

Kursen syftar till att studenten utvecklar kunskap och färdigheter inom grundläggande programmering.

Under kursens gång inhämtas erfarenheter av problemanalys, problemlösning, implementering och dokumentation av programvara.

Kursens innehåll

- Kursen fokuserar på praktiska **programmeringsfördigheter**, förankrade i relevant teori. Utgångspunkten är studentens arbete med **specifika problem**. Kursen behandlar:
 - **problemlösningsstrategier**
 - användning av språkkonstruktioner från **strukturerad programmering**
 - **implementering** och utvärdering av egna program

Lärandemål – Kunskap och förståelse

- kunna visa förståelse för **grundläggande begrepp** inom strukturerad **programmering**
- visa ett **algoritmiskt tankesätt** och förståelse för att använda informell logik vid utformandet av algoritmer
- kunna övergripande diskutera **skillnader och likheter mellan strukturerad och objektorienterad programering**

Lärandemål – Färdighet och förmåga

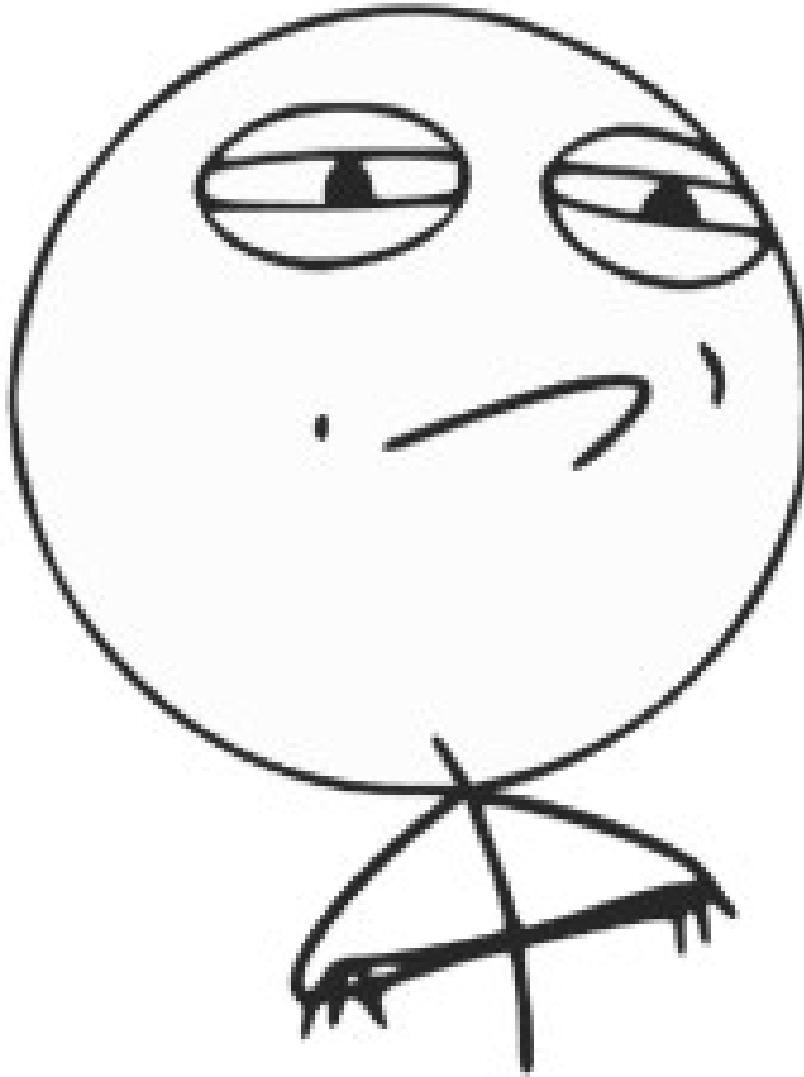
- kunna strukturera och **implementera enklare program** i ett programmeringsspråk
- självständigt från en **problembeskrivning** kunna **analysera** problem och **konstruera** ett fungerande program
- kunna motivera sin **programkod** med avseende på lämpliga **kontrollstrukturer, läsbarhet** samt **konventioner** i aktuellt programmeringsspråk
- kunna använda **tredjeparts- och standardbibliotek** vid programmering

Lärandemål – Värderingsförmåga och förhållningssätt

- kunna **kritiskt resonera** kring alternativa lösningar för ett givet programmeringsproblem

Problemlösning

CHALLENGE ACCEPTED



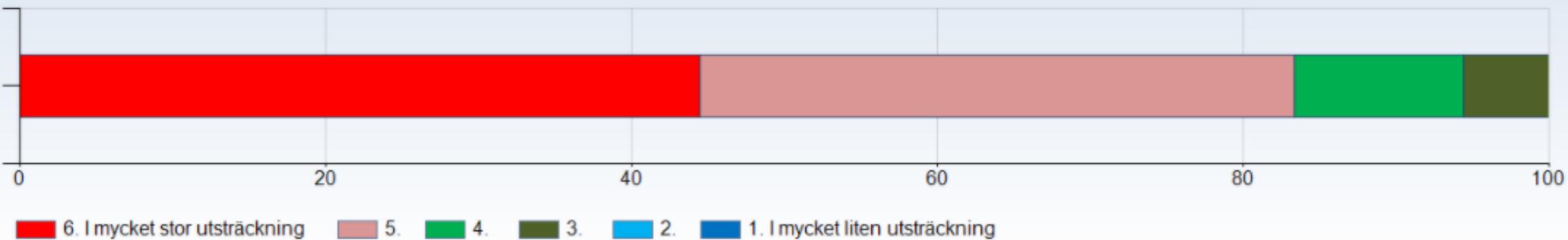
Kursutvärdering

HT2019

I vilken utsträckning anser du dig ha uppnått kursens lärandemål?

Lärandemål

Lärandemål	Antal svar
1. I mycket liten utsträckning	0 (0%)
2.	0 (0%)
3.	1 (6%)
4.	2 (11%)
5.	7 (39%)
6. I mycket stor utsträckning	8 (44%)
Summa	18 (100%)



Lärandemål	Medelvärde	Standardavvikelse
	5,2	0,9

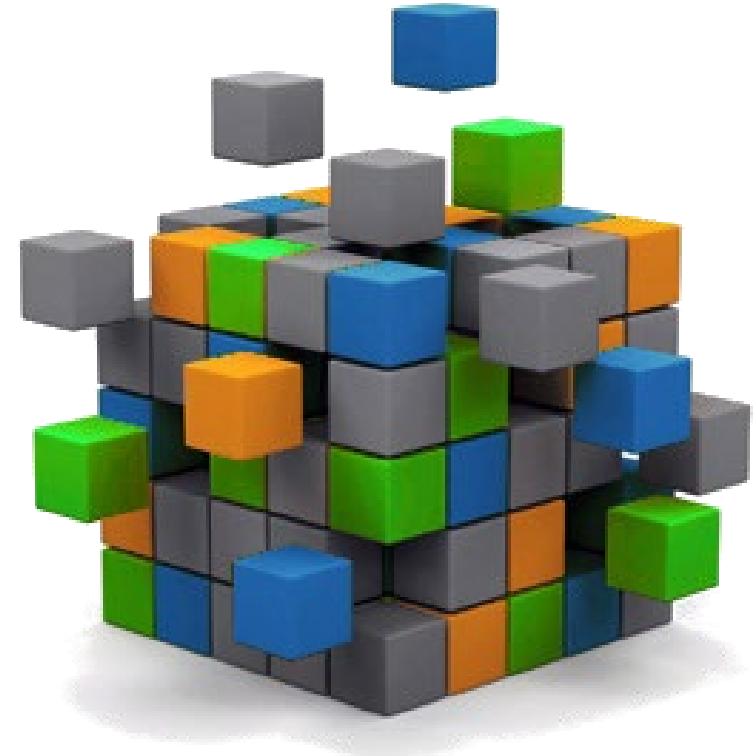
Huvudpunkter från förra kursutvärderingen

- 1) Studenterna var generellt nöjda med kursen
- 2) Studenterna tyckte att examinationsformerna var bra
- 3) Studenterna tyckte att föreläsningarna var bra
- 4) Studenterna tyckte att laborationerna var bra

- 5) Studenterna tyckte att just inlämningsuppgift 3 var svår
 - 1) Vi har därför gjort denna till en gruppuppgift i år
- 6) Studenterna ville ha mer laborationstid
 - 1) Antalet laborationstimmar är dubblade denna version av kurserna

Kursens moduler

1. Introduktion till Python
2. Funktioner
3. Iteration och selektion
4. Listor och lexikon
5. Fil- och felhantering
6. Webbapplikationer



En modul per vecka, en inlämningsuppgift per vecka

Varje modul har..

- Föreläsningar
- Laborationer
- Inlämningsuppgifter
 - med undantag för första modulen



Modulen börjar



Onsdag	18/11	13-15	F: Funktioner i Python	Anton	Zoom	
Fredag	20/11	09-12	F+L: Funktioner i Python	Anton	Zoom + Discord	
Måndag	48	23/11	13-17	L: Funktioner i Python	Amanda	Discord
Tisdag		24/11	08-12	L: Funktioner i Python	Anton	Discord

2

Modulen avslutas

Examinerande moment

- Inlämningsuppgifter
- Deadline: söndagar varje vecka från vecka 48.

1. 29/11

2. 6/12

3. 13/12

4. 20/12

5. 10/1 < OBS. efter julen



Kurslitteratur

- Downey, Allen B. (2015) Think Python, O'Reilly, Beijing, ISBN-13: 9781491939369
- <http://greenteapress.com/wp/think-python-2e/>



da354a.ia-mau.se
Vår kurswebbplats

Kursöversikt

Planering

Kursplan →

Canvas →

1. INTRODUKTION TILL PYTHON ^

Översikt

Föreläsning

Ö.1: Python & Idle

Ö.2: Variabler & datatyper

Ö.3: Debugging & kommentarer

Ö.4: Case (1) - Valutakonverterare

Ö.5: Case (2) - Budgetberäknare

Ö.6: Case (3) - Valfritt

Introduktion till programmering (DA354A)

Denna webbplats innehåller material från en grundkurs i webbutveckling. De som läser [kursen](#) är den primära målgruppen, men mycket kan förhoppningsvis vara användbart även för andra. Allt material är publicerat under en [fri licens](#), vilket gör att det kan återanvändas av andra, så länge upphovsman anges.

Kurslitteratur i denna kurs är boken *Think Python* av Allen B. Downey. Ni kan den finna gratis på nätet antingen som [webbsida](#) eller som [PDF](#). Föredrar du en fysisk kopia av boken kan du hitta en återförsäljare [här](#).

Syfte

I den bakomliggande kursplanen beskrivs syfte och mål:

Kursen syftar till att studenten utvecklar kunskap och färdigheter inom grundläggande programmering. Under kursens gång inhämtas erfarenheter av problemanalys, problemlösning, implementering och dokumentation av programvara.

I denna kurs innebär det programmering i språket [Python](#). Kursen är uppdelad i 7 st moduler som progressivt bygger på varandra. Modulerna kommer att publiceras löpande under kursens gång, och en kort presenteras av modulerna hittar ni nedan.

1. Introduktion till Python

Modulen fokuserar på att komma igång med programmering i Python. Det innebär bl.a. att installera den mjukvara som behövs för att skriva och köra Python-kod, samt de mest grundläggande byggnesterna i programmering som t.ex. *variabler, datatyper & utskrift av data*.

2. Funktioner

Modulen fokuserar på *vad* funktioner är för något, *varför* vi vill använda funktioner, samt *hur* vi skriver och använder funktioner i Python. Vi tittar även på *argument* och *parametrar* som gör att våra funktioner kan användas i mer generellt syfte, beroende på vilken data som skickas till funktionen när den körs.

3. Iteration och selektion

[Kursöversikt](#)[Planering](#)[Kursplan](#)[Canvas](#)[1. INTRODUKTION TILL PYTHON](#)[Översikt](#)[Föreläsning](#)[Ö.1: Python & Idle](#)[Ö.2: Variabler & datatyper](#)[Ö.3: Debugging & kommentarer](#)[Ö.4: Case \(1\) - Valutakonverterare](#)[Ö.5: Case \(2\) - Budgetberäknare](#)[Ö.6: Case \(3\) - Valfritt](#)

Planering

All undervisning i kursen sker i digital form i höst p.g.a. *covid-19*. All information om vilket Zoom-rum föreläsningarna kommer att ske i, eller hur man bärst får hjälp på laborationerna i Discord kommer att presenteras på kursplats på Canvas.

Varje modul är **en vecka** och introduceras varje **onsdag**. Alltså börjar den första modulen onsdagen den 11/11, den andra modulen den 18/11, den tredje moduel den 25/11, osv.

Förkortning	Innebörd
F	Föreläsning
L	Laboration/övningar
F+L	Föreläsning & Laboration/övningar

Vecka	Datum	Tid	Moment	Lärare/Assistent	Plats	Modul
46	10/11	10-12	Kursintroduktion	Anton	Zoom	
	11/11	13-15	F: Introduktion till programmering	Anton	Zoom	
47	16/11	13-17	L: Introduktion till programmering	Amanda	Discord	1
	17/11	08-12	L: Introduktion till programmering	Anton	Discord	
	18/11	13-15	F: Funktioner i Python	Anton	Zoom	
	20/11	09-12	F+L: Funktioner i Python	Anton	Zoom + Discord	2
48	23/11	13-17	L: Funktioner i Python	Amanda	Discord	
	24/11	08-12	L: Funktioner i Python	Anton	Discord	
	25/11	13-15	F: If-satser & Loopar	Anton	Zoom	

[Kursöversikt](#)[Planering](#)[Kursplan](#) ↗[Canvas](#) ↗[1. INTRODUKTION TILL PYTHON](#)

^

[Översikt](#)[Föreläsning](#)[Ö.1: Python & Idle](#)[Ö.2: Variabler & datatyper](#)[Ö.3: Debugging & kommentarer](#)[Ö.4: Case \(1\) - Valutakonverterare](#)[Ö.5: Case \(2\) - Budgetberäknare](#)[Ö.6: Case \(3\) - Valfritt](#)

Modul 1 - Introduktion till Python

Översikt av modul 1

I den första modulen kommer vi att titta på några grundläggande koncept & byggstenar i programmering. Vi kommer framförallt att titta på:

- Variabler
- Datatyper
- Utskrifter
- Indata (från användaren)

Denna modul syftar till att introducera många av dessa koncept, i ett (för många) nytt programspråk, *Python*. Ö i tabell nedan, och menyn till vänster är förkortning för **övning**.

Innehåll	Kommentar
Föreläsning	<i>Publiceras senare</i>
Ö.1: Python & Idle	En introduktion till språket Python och miljön Idle där vi skriver vår kod
Ö.2: Variabler & datatyper	En introduktion till grundläggande byggstenar i programmering
Ö.3: Debugging & kommentarer	En introduktion till att hitta (och åtgärda) fel i kod, samt kommentera kod
Ö.4: Case (1) - Valutakonverterare	En något större uppgift att träna på
Ö.5: Case (2) - Budgetberäknare	En något större uppgift att träna på
Ö.6: Case (3) - Valfritt	En något större uppgift att träna på
Inlämningsuppgift	<i>Ingen uppgift i den första modulen</i>

Rekommenderad läsning

→ [Think Python - Chapter 1 The way of the program](#)

Frågor?

da354a.ia-mau.se
Vår kurswebbplats