

`</>`

Intro

"Vad gör vi när data strukturen inte är hållbar?"

Ta upp



- Handuppräckning
- Vart/när Material Publiceras
- Inspelning
- Planering & Grupparbete
- Feedback ska ske inom rätt forum
- Kontakt & Förväntningar

	Måndag	Tisdag	Torsdag
Vecka 5	Intro, versionshantering & Terminal	Webbkomunikation & Protokoll	FastAPI: XML, JSON, API, CSV
Vecka 6	Workshop - Pair Programming Workshop Github Projects, kanban, MVP, Merge Conflicts, pull requests & Issues	Pandas & Databas & CSV, XML, JSON	Pipelines: ETL vs ELT
Vecka 7	Workshop - Rensning, validering, transformering av Data	Pandas - LABB INTRO (analys-projekt)	Workshop AGILT & Eget Arbete
Vecka 8	EGET ARBETE	EGET ARBETE	DEADLINE
Vecka 9	Workshop - Docker Basics		Kafka & RabbitMQ - Project Intro
Vecka 10	EGET ARBETE		Sprint 1 - demo
Vecka 11	EGET ARBETE		Sprint 2 - demo
Vecka 12		EGET ARBETE	Sprint 3 - demo
Vecka 13	EGET ARBETE		DEADLINE - Muntlig presentation
Resurser			
Tider	Måndag	9-16 (Heldag)	
	Tisdag (v5-8)	9-16 (Halvdag)	
	Torsdag	9-16 (Heldag)	

Table of Contents

01

Kursplan &
Översikt

03

02

Intro, Terminal

04

Uppgifter
&
Övningar

KURSPLAN

Innehåll

Syftet med kursen är att ge den studerande färdigheter i programmering av dataplattformar, ETL pipelines och verktyg för dataintegrering. Den studerande får färdigheter och kunskaper i branschen vanligt förekommande programmeringsspråk.

Source: *Kursplan (Learnpoint)*

KURSPLAN

Utbildningsmoment:

- Dataplattformar, bakgrund och syfte
- Git och github i teamkontext
- Komponenter och teknologier i en data plattform
- ETL vs ELT
- Utveckling av mjukvara mot databaser
- Använda Python mot relationsdatabaser och andra datakällor såsom csv, http xml/json
- Använda Python mot realtidsdataströmmar såsom message queues och/eller event streaming platforms
- Använda Python och för att rensa, validera och transformera data
- Workflow processer

Kunskap

ID	Beskrivning
KUN9	Redogöra för typiska komponenter och teknologier i en dataplattform
KUN10	Redogöra för begreppet ETL pipelines

Färdigheter

ID	Beskrivning
F10	Med hjälp av Python kunna läsa, ändra och skapa data i olika dataformat och databaser
F11	Kunna skapa program som transformerar och strömmar data genom programmering i Python
F12	Kunna jobba agilt i grupp med git och github

Kompetens

ID	Beskrivning
KOM6	Självständigt kunna läsa och skriva till datakällor som realtidsdataströmmar

Former för kunskapskontroll

ID	Format för kunskapskontroll	Betygsskala
KUN9; KUN10, F10	Inlämningsuppgift	IG/G
F11,F12,KOM6	Projektarbete	IG/G/VG <i>För att få betyget Väl Godkänt (VG) ska studerande planera, utveckla och leverera ett projekt inom data platform development som uppvisar hög teknisk kvalitet, kreativitet i lösningsförslag, och reflektion kring val av teknologier och metodik. Arbetet ska också visa att studerande kan hantera komplexa kodmässiga utmaningar med god struktur, tydlig dokumentation och med hänsyn till branschens krav.</i>

Overview



Moduler:

- Python Basics - Repetition
- Terminalen
- Git
- Versionshantering



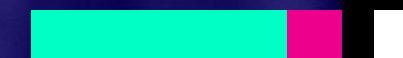
Utbildningsmoment

- **Dataplatfformar, bakgrund och syfte**
- **Git och github i teamkontext** ✓
- **Komponenter och teknologier i en data platform**
- **ETL vs ELT**
- **Utveckling av mjukvara mot databaser**
- **Använda Python mot relationsdatabaser och andra datakällor såsom csv, http xml/json**
- **Använda Python mot realtidsdataströmmar såsom message queues och/eller event streaming platforms**
- **Använda Python och för att rensa, validera och transformera data** ✓
- **Workflow processer**

02

Intro, Terminal & Versionshantering

Intro Data Platform



Talk LAB



3 min

Diskutera:

Vad tänker du på när du hör
'DATA PLATFORM'

Vad är '**data**'?

Hur **transporteras data över nätverk**?

Data Platform

(Vad är det?)

En dataplattform är teknik som samlar in, lagrar och bearbetar data så att den är korrekt, tillgänglig och pålitlig att använda

Data Platform

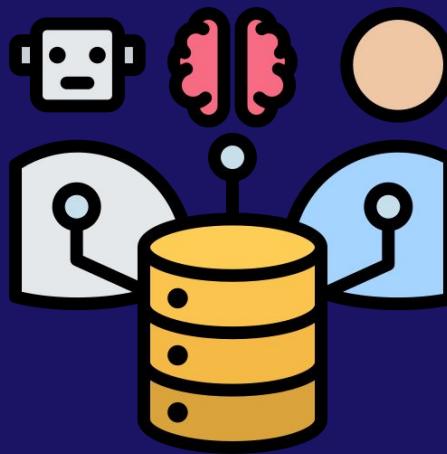
(Är det en app, databas?)

En dataplattform är ett parallellt system som kopierar data från andra system för att man inte ska störa dem... och för att kunna arbeta med data fritt

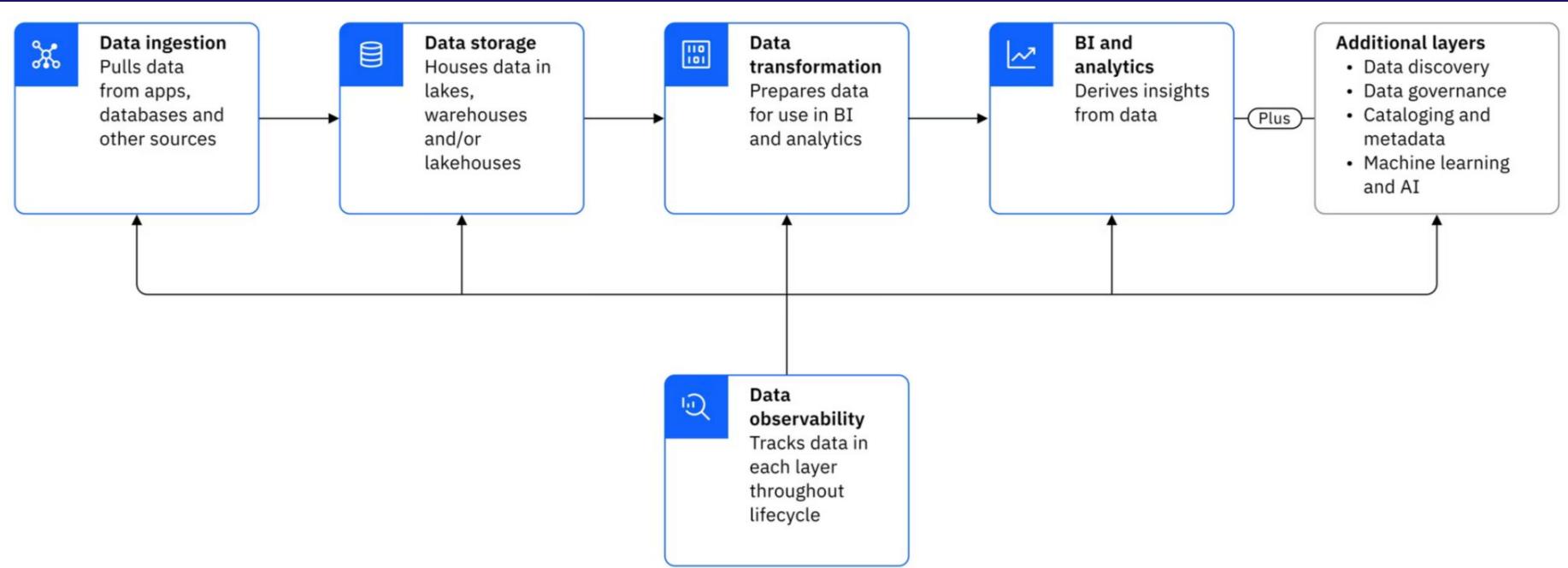
Data Platform (Typer?)

- Enterprise data platform (EDP)
- Big data platform (BDP)
- Cloud data platform (CDP)
- Customer data platform (CDP)

Läs mer: <https://www.ibm.com/think/topics/data-platform>

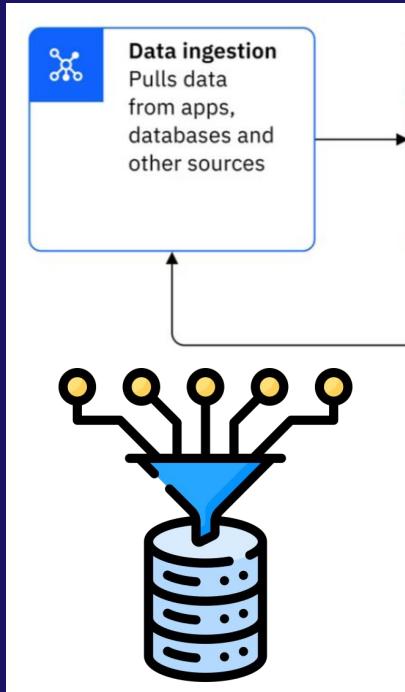


Data Platform består av... (Lager)



Data Ingestion

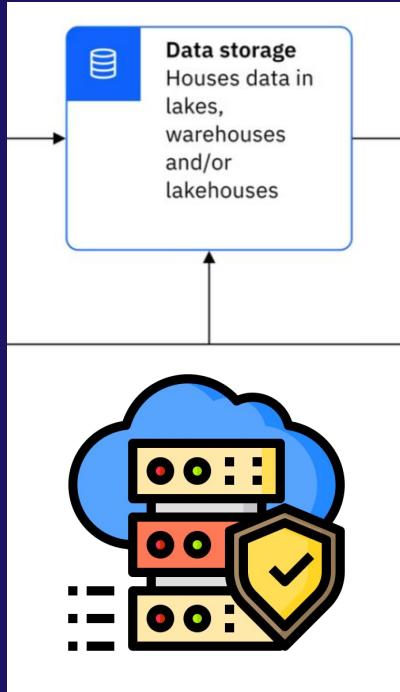
(Hur data kommer in)



Att hämta data från **system, filer eller API:er** in i plattformen.

Data Storage

(Vart data sparar ner)



Data warehouse (*kan vara SQL/noSQL databaser*)

Lagrar strukturerad data som är färdig för analys och rapporter.

Data lake

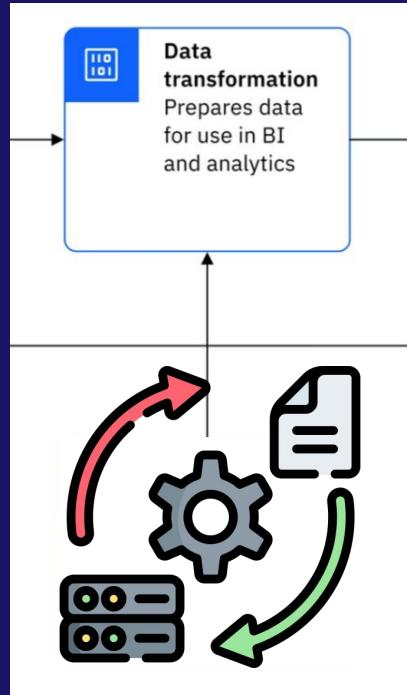
Lagrar stora mängder rådata i olika format (*ostrukturerad och strukturerad*).

Data lakehouse

En kombination av data lake och data warehouse.. flexibel lagring men med bättre struktur och prestanda.

Data Transformation

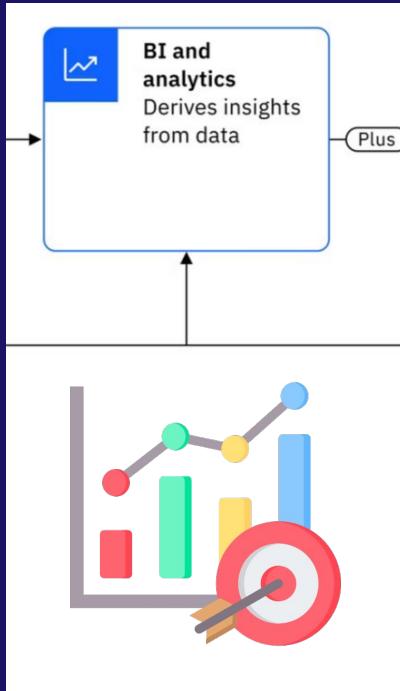
(Hur data bearbetas)



Data transformation

Att göra om rådata till något användbart.
Rensa, sortera, slå ihop och anpassa data.

Business Intelligence & Analytics (Vart datan sparas)

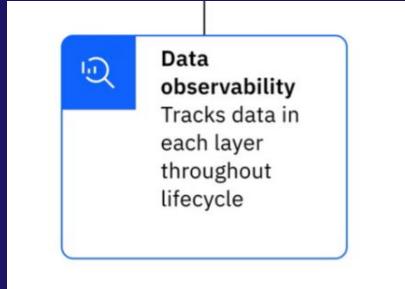


BI & Analytics

Business Intelligence & Analytics handlar om att samla och analysera data som lagras i exempelvis **databaser, data warehouses eller data lakes** för att skapa insikter och beslutsunderlag

Data Observability

(Data översikt, Flöde & Problem)



Data observability

Att ha koll på att datan funkar som den ska.

Upptäcka om data saknas, är fel eller kommer för sent.

IBM & Generative A.I

(Data & Framtiden)

Ni kommer höra mycket om generativ AI, men det viktiga här är inte själva AI:n.
Det viktiga är **datan bakom**.

Företag som vill använda AI måste först ha koll på sin data:

- var den finns
- om den är korrekt
- om den går att hämta

Det är det en dataplattform handlar om då AI bara är en konsument av datan!

Video: <https://www.ibm.com/think/videos/ai-academy/generative-ai-for-data-management>



Talk LAB



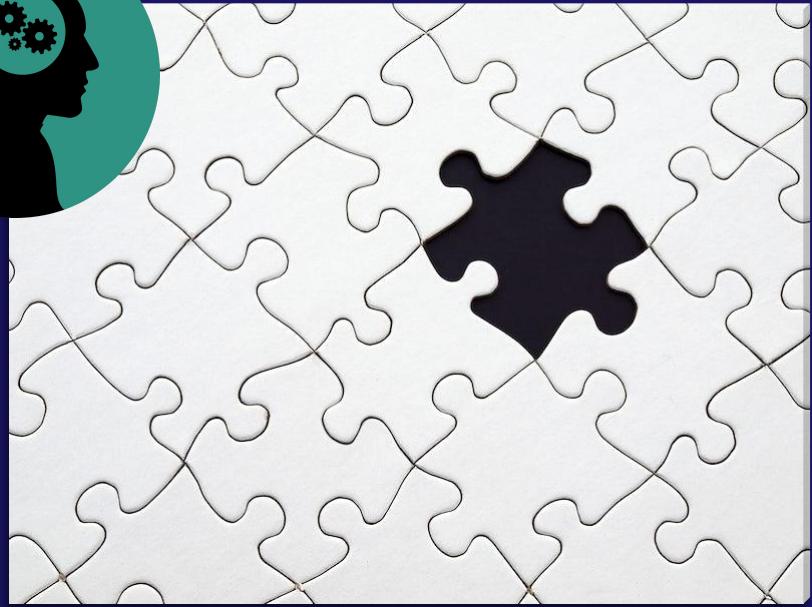
3 min

Diskutera:

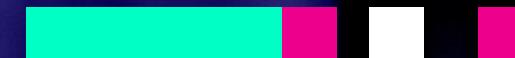
Är **Data Platform** enbart för A.I?

Vad händer om **data** inte går att hämta inom ett system?

Frågor?



Data Overview



Fundering

*Hur många exempel kan du komma på
gällande data som sparas ner?*



Talk LAB



2 min

Diskutera:

Brainstorm: all form av data
ni kan komma på..

GRUPP

Vem du interagerar mest med – och hur ofta

Hur dina vanor påverkas av andra i din närhet (t.ex. när du anpassar dig efter någon annans Spotify eller Netflix)

Om du verkar vara en ledare eller följare i gruppbeende online

Hur ofta du gillar vs bara tittar (tyst vs aktiv användare)

Vilken typ av enhet du använder (mobilmodell, operativsystem, webbläsare)

Ditt batterinivåbeteende (t.ex. om du ofta har låg batteri – det kan indikera livsstil)

Din skärmstorlek, tangentbordslayout och språk

Om du använder hörlurar eller högtalare

Vilka enheter är kopplade till ditt wifi-nätverk (smartahem-prylar)

Exakt plats i realtid

(GPS-spårning – "du var på ICA klockan 17:34")

Vem du har träffat

(Bluetooth-spårning mellan mobilier)

Vad du lyssnar på och när

(Spotify vet att du lyssnar på deppig musik på kvällarna)

Vad du skrivit men aldrig skickat

(texter du raderat eller inte publicerat)

Dina sökningar

(även de pinsamma eller känsliga)

När du sover och vaknar

(mobilens rörelsemönster + skärmanvändning)

Vilka appar du rör dig mellan – och hur ofta

Hur snabbt du skriver, hur du rör musen

(beteendemönster för identitet)

Vad du lagt i en varukorg men inte köpt

Hur ofta du besöker en webbshop innan du handlar

Dina returer, avbokningar eller om du bara "fönstershoppar"

Vilken typ av rabatter du klickar på (dvs vad som triggas dig att handla)

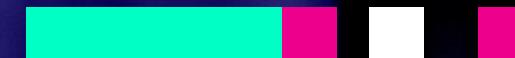
Vad du stannar upp och tittar på, även om du inte klickar

Hur länge du tittar på en bild eller ett inlägg

Vilken typ av innehåll får dig att scrolla vidare eller pausa

Vilka färger, ord eller teman du reagerar mest på

What is data? Binary



Binary (Explained)

Det binära talsystemet är grundat på talbasen 2 och utnyttjar endast **två siffror**, vanligen 0 och 1.

Liksom i det decimala talsystemet är siffran längst till höger minst signifikant. Med en siffra kan endast talen 0 och 1 skrivas



Binary (Strengths)



Styrka:

- Kan tolkas enkelt av system
- Förhindra korruption
- Säkerställ korrekt data

*"Nu kan vi skicka datan och
förhindra att data försvinner"*

Protocol (Webbommunikation)

0 1 0 1 1 1
0 1 0 0 1 0
0 1 0

"Hur skickar vi/tar
emot data?"

"Säkerhet?"

"Nätverk då?"



Data & Programming

(What do we have in common?)





Problem

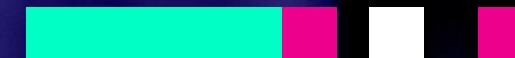
What is data?



Solution

Data can be summed up as Binary (0's and 1's) where we store information. Thanks to binary we can make sure that data is received correctly!

Data Types Overview



Primitive Data types

Text Type: `str`

Numeric Types: `int`, `float`, `complex`

Sequence Types: `list`, `tuple`, `range`

Mapping Type: `dict`

Set Types: `set`, `frozenset`

Boolean Type: `bool`

Binary Types: `bytes`, `bytearray`, `memoryview`

None Type: `NoneType`

Data type

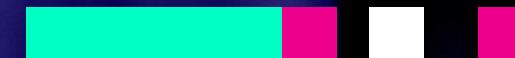
Tidsrelaterat

- Datum
- Tid
 - a. Timestamp
 - b. Timezone

Varför Data typer?

- Lagring (*minne, disk, nätverk*)
- Prestanda
- Validering & kvalitet (*Missing / no data*)
- Tolkning (*vad betyder värdet, hur tolkas "2" mot 2?*)

Open Data Overview



Consuming Data

Consuming data in programming involves accessing and utilizing information from external sources, often through APIs.

Open data consumption empowers developers to create innovative solutions, enhance functionality, and provide meaningful insights by connecting users with a wealth of accessible, structured information

1.6m

1.6 million datasets across EU countries

(A **dataset** is a structured collection of data, often organized in rows and columns,
similar to a table in a database or a spreadsheet.)

Terminologies

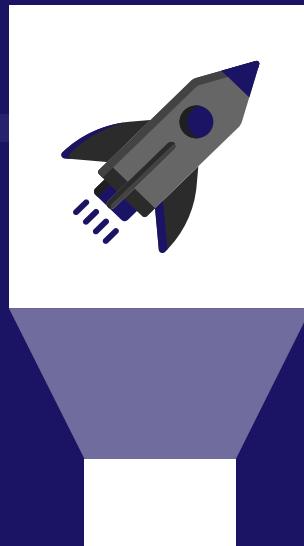
Dataset

A structured collection of related data.



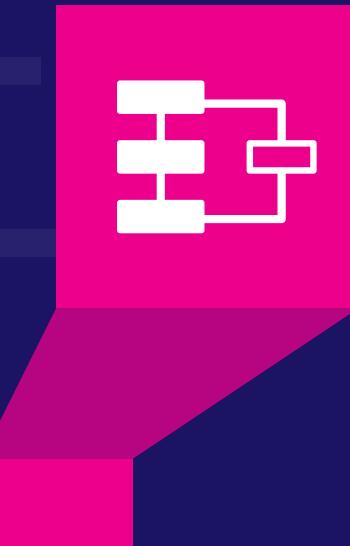
API/WS

A set of protocols for accessing and interacting with data or services

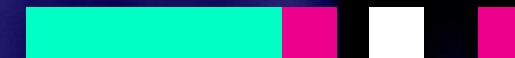


Open Data

Publicly available data that anyone can access, use, and share.



Terminal CLI-gränssnitt

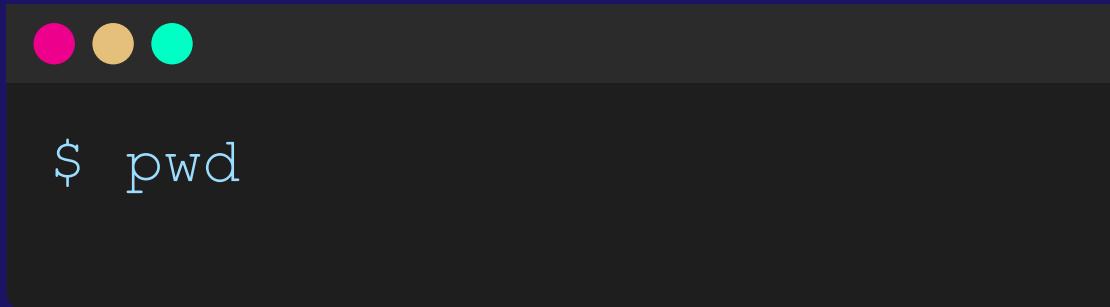




How to move,
inspect, and control
your system

Terminal

(Get Current Path)



A screenshot of a macOS terminal window. The window has a dark gray background and three colored circular icons in the top-left corner (red, yellow, green). In the main pane, the command '\$ pwd' is typed in white text.

```
$ pwd
```

Where am I?

Terminal

(List Content)

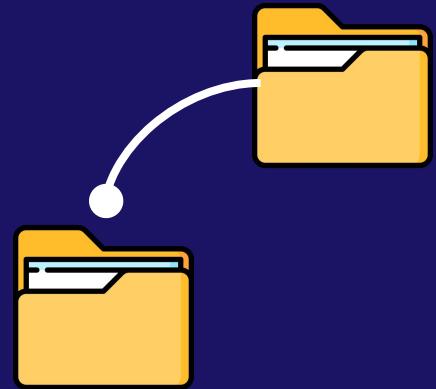
```
$ ls      # MAC / LINUX  
$ dir     # WINDOWS
```

See the contents of a directory



Terminal (Navigate)

```
$ cd FOLDER-NAME  
$ cd ..      # Navigate back up
```



Move from one dictionary -> another

Terminal (Clear)

```
$ clear  
# CTRL + L (same effect)
```

Clear the terminal from text



Work Lab

Terminal



- Öppna Terminal
- Testa:
 - \$ pwd
 - \$ ls / dir
 - \$ cd

03

Versionshantering

Git & Github Versionshantering



100m

Github Användare

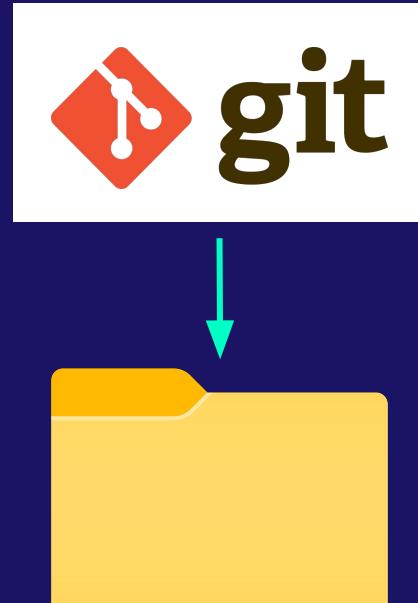
Över 400 miljoner

Git & GitHub

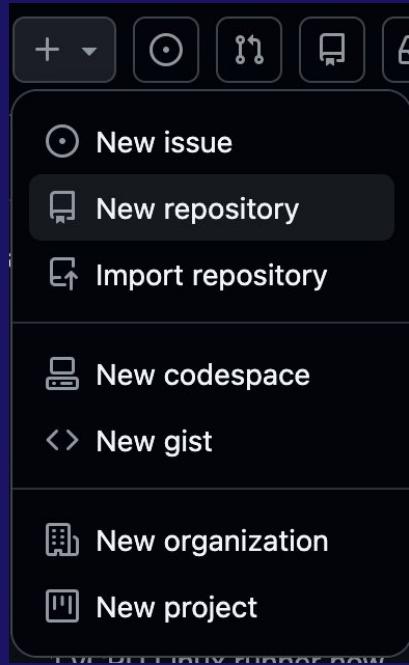
(Create Git Project)

```
$ git init
```

Generate new Github project at current directory



Git & GitHub (Create Repository)



Name your Repository (free choice)

Choose Repository visibility:

- Public (anyone can see it)
- Private (only you and collaborators)

Leave "Add a README" unchecked (we want an empty repo to push to)

Click Create repository

Git & GitHub

(Copy Remote)

...or create a new repository on the command line

```
echo "# testtest" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Krillinator/testtest.git
git push -u origin main
```

Copy the URL of your Repository, ignore the rest

Git & GitHub

(Connect Remote)

```
$ git remote add URL
```

*Connect your local project with Online Repository.
URL == Repository URL*



Git & GitHub

(Version Control Files & Directories)

```
$ git add .
```

*(from where we stand)
Add ALL files & folders -> Git*



Git & GitHub

(Save with Message)

```
$ git commit -m ""
```

Save your work (locally) with a message

"Add: New Feature"



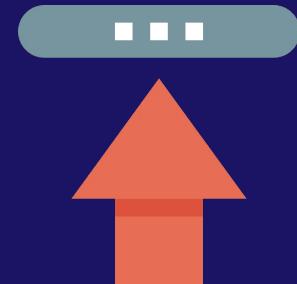
Git & GitHub (Publish Project)

```
$ git push -u origin main
```

Push Changes to: main (branch)

Only once per branch:

\$ -u origin BRANCH-NAME



Work Lab

Git



- Skapa Repository
- Skapa nytt Python projekt
 - \$ git init
 - \$ git add remote URL

Don't do anything else

Git status

\$ git status

Vad var jag...? (Förklarat)

```
$ git init
```

```
???
```

*Glömt vad som gjorts?
STATUS av git?*

```
$ git add .
```

```
$ git push
```

Lösning (git command)

```
$ git status
```

*What's the current git status?
Commits? Everything up-to-date?*



Work Lab

Git



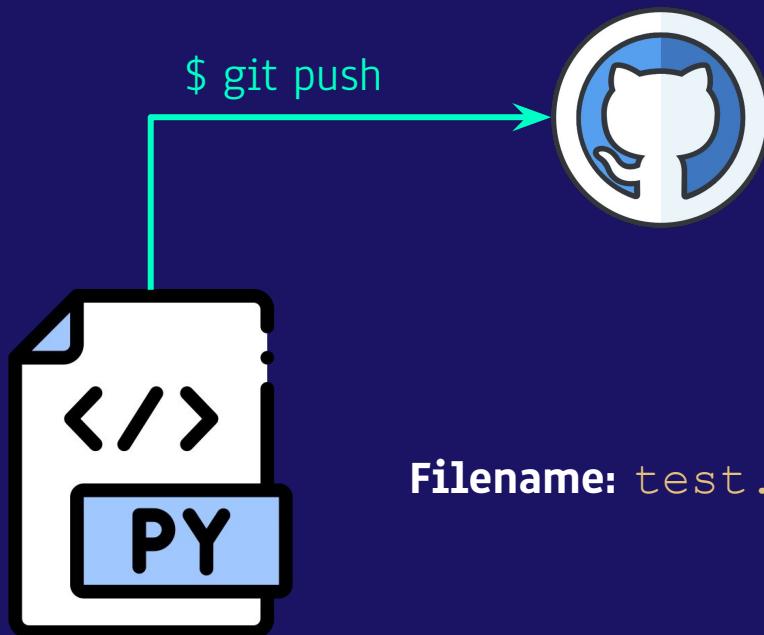
Inom samma projekt

- \$ git status
- \$ git add .
- \$ git status
- \$ git commit -m ""
- \$ git status
- \$ git push -u origin main

Git List Files

\$ git ls-files

The problem (Förklarat)



The problem #2 (Förklarat)

1 fil lokalt

1 fil på molnet

Allt är som det ska



Filename: test.py

The problem #3 (Förklarat)

Vad händer om vi **ändrar filnamnet lokalt** och gör en
\$ git push



Filename: test2.py

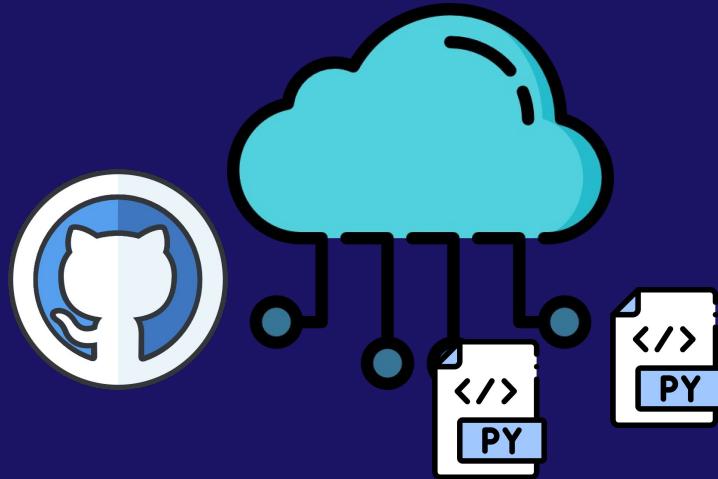


Filename: test.py

The problem #4 (Förklarat)

Dublett skapad...

Följdfråga: hur tar vi bort filer som ligger uppe på GitHub?



Filename: test.py

Filename: test2.py

Lösning (Identifiera)

Leta efter filen med:

```
$ git ls-files
```

Filtrera sökresultat med

```
$ git ls-files | grep NAMN
```

```
terminal,-git-&-github/.DS_Store
terminal,-git-&-github/branching/branches-basics.html
terminal,-git-&-github/branching/git-branch-flow-diagram.html
terminal,-git-&-github/branching/git-branch-simplified-flow-diagram.html
terminal,-git-&-github/git-&-github-basics/git-&-github-common-issues.html
terminal,-git-&-github/git-&-github-basics/git-&-github-first-project.html
terminal,-git-&-github/git-&-github-basics/git-flow-diagram-tbd.html
terminal,-git-&-github/git-&-github-basics/github-account-creation.html
terminal,-git-&-github/git-&-github-basics/github-password-token.html
terminal,-git-&-github/git-&-github-basics/github-removing-files-from.html
terminal,-git-&-github/pull-request/cheatsheet-pull-requests.html
terminal,-git-&-github/pull-request/pull-request-flow-diagram.html
terminal,-git-&-github/terminal-basics/sudo-superuser-do.html
terminal,-git-&-github/terminal-basics/terminal-&-files-basics.html
terminal,-git-&-github/terminal-basics/terminal-flow-diagram.html
terminal,-git-&-github/terminal-basics/terminal-get-started.html
terminal,-git-&-github/user-&-technical-stories/user-&-technical-stories.html
terminal,-git-&-github/user-&-technical-stories/user-stories-vs-technical-fundamentals.html
```

Många filer...

Filtrera med 'grep'

To abort git file (git remove)

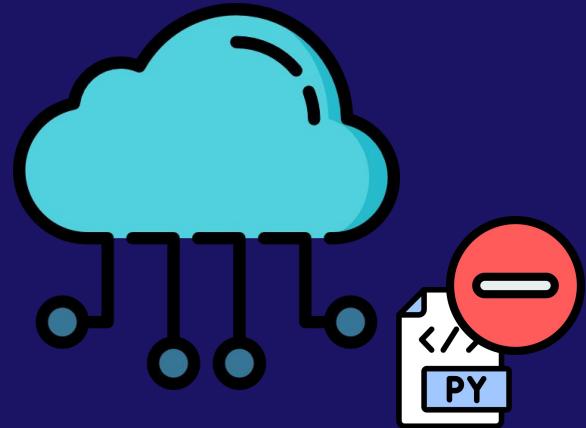


```
$ git rm FILENAME
```

*Git remove python_file.py
Then you can proceed to:*

```
$ git add .  
$ git commit -m ""  
$ git push
```

And it'll disappear from the cloud



Work Lab

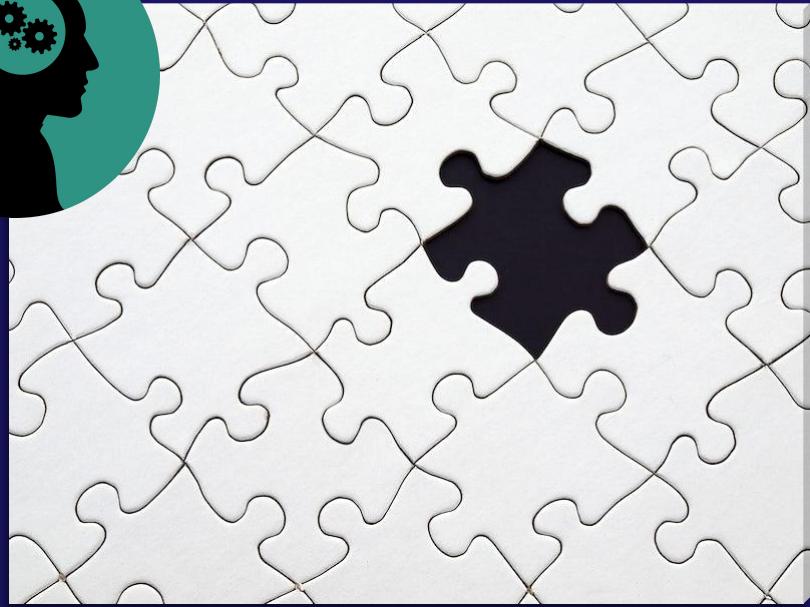
Git



Inom samma projekt

- \$ git status
- \$ git add .
- \$ git status
- \$ git commit -m ""
- \$ git status
- \$ git push -u origin main

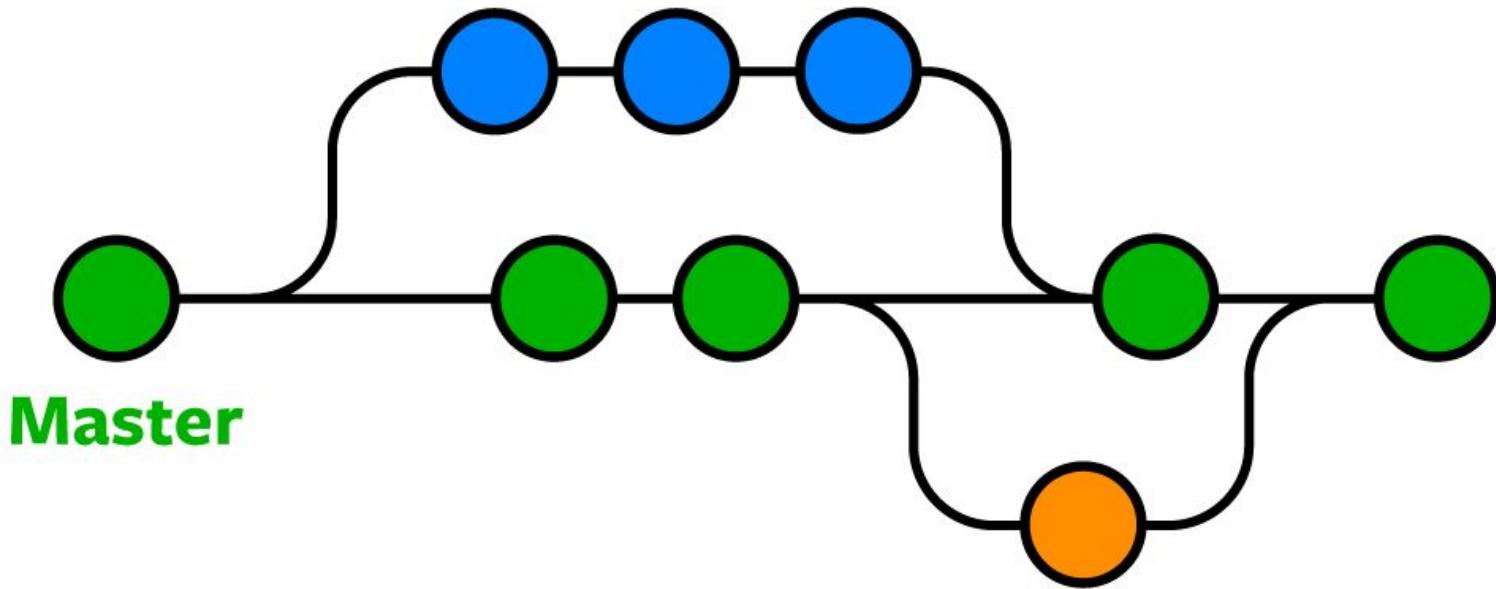
Frågor?



Git Branch

\$ git branching

Your Work



Master

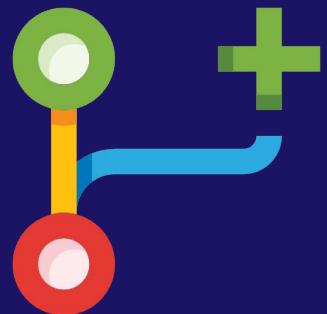
Someone Else's Work

Git Branching

(create new branch)



```
$ git checkout -b NAME
```



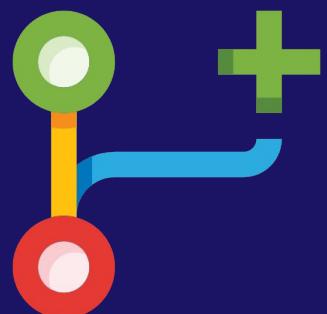
Create AND switch to branch with NAME

This assumes your current branch has no uncommitted work on it. Git commit if necessary.

Git Branching

(create new branch)

```
$ git branch -r
```



Check out branches on the remote (cloud)

Local PC

GitHub

Clone Repository

`$ git clone URL`

Repository

Create Branch

`$ git checkout -b NAME`

Commit Work

`$ git commit -m "TITLE" -m "Description"`

Push Branch

Branch Available

Fetch

`$ git fetch`

Others can now see branch

Guide: How to: About Git Branches

Work Lab

Git



Inom samma projekt

- \$ git branch
- \$ git checkout -b NAME
- \$ git branch
- \$ git add .
- \$ git commit -m ""
- \$ git push -u origin
BRANCH-NAME

04

Uppgifter

&

Eget Arbete

Uppgifter

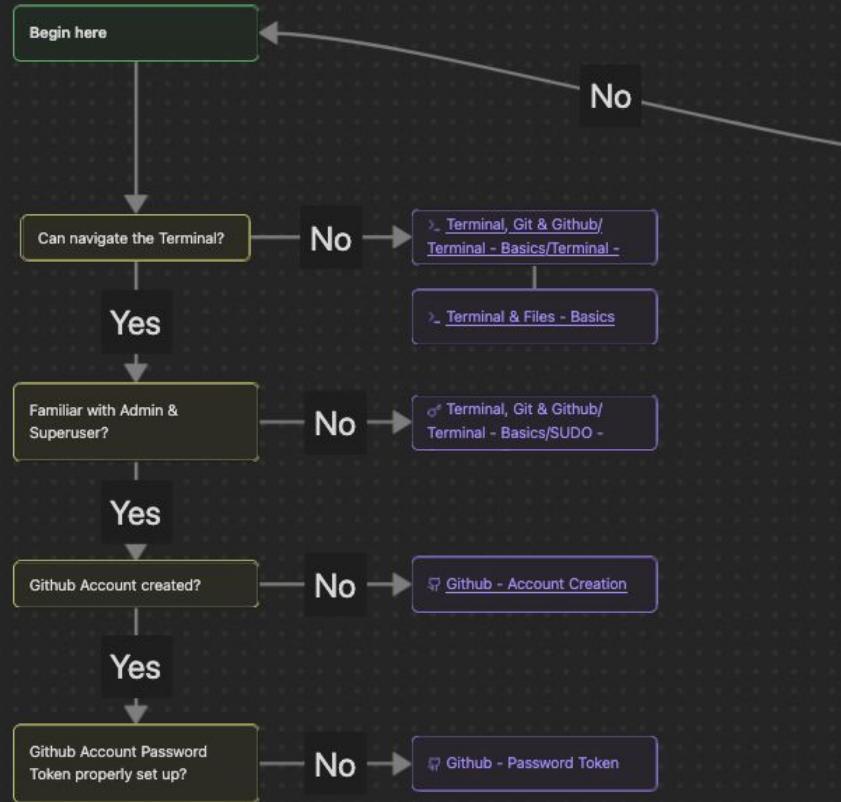
Välkommen till första uppgiften!

Uppifterna är till för att testa dina färdigheter och kunskaper för att både öva och repetera på det vi har arbetat med under föreläsningarna.

Dessa är **INTE** obligatoriska.
Men är ämnen ni kommer testas mot.



Zoom in with SCROLL



Get Started!

(Learn the Basics)

Get Started here: [Developer Fundamentals](#)

NOTE: Scroll to Zoom!

Image will be small upon arrival

Terminal (Navigation)

What is the Terminal?

The Terminal (sometimes called *command line* or *shell*) is a way to **control your computer with text commands** instead of clicking.

It lets you **navigate, create files/folders, and run programs** directly.

Think of it as:

"Talking to your computer in its own language, step by step."



Guide: [How to: Terminal - basics](#)



Terminal

(Files & Directories)

Working with Files in the Terminal

This section focuses on **creating, editing, reading, and filtering files** directly from the command line.. without leaving the terminal.



Guide: [How to: Create Files & Folders in Terminal](#)

Terminal (bonus) (Admin & Privileges)



SUDO - Superuser Do

Guide: [How to: SUDO](#)



GitHub Token (bonus)

(Token based authentication)

What is a GitHub Token?

A Personal Access Token (PAT) is a secure alternative to your password when pushing or pulling code from GitHub.

Since August 2021, GitHub no longer allows normal account passwords for Git operations.

Think of it as:

"Your personal access key that Git uses instead of your GitHub password."



Guide: [How to: Password Token](#)

Github, Git & Repository (First Project)

What is Git & GitHub?

Git is a version control system.. it lets you **track changes** in your code over time.

GitHub is an online platform where you can **store Git repositories**, collaborate, and share your projects.

Think of it as:

"Git saves your project's history, GitHub makes it easy to share that history with others."



Guide: How to: First Project with Git



Github, Git & Repository (First Project)

What is Git & GitHub?

Git is a version control system.. it lets you **track changes** in your code over time.

GitHub is an online platform where you can **store Git repositories**, collaborate, and share your projects.

Think of it as:

"Git saves your project's history, GitHub makes it easy to share that history with others."



Guide: [How to: About Git Branches](#)



Binary & ASCII Bonus (Valfritt)



Binary (Decimal Calculation)

ASCII är en encoder på **128 symboler**.

Låt oss **göra om "Hi"** till **binär data**

Eftersom att det bara finns två tal 0 1, så används **power of 2^x**

H -> 01001000

i -> 01101001

EXAMPLE: Decimal Calculation for H

Binary Data:	0	1	0	0	1	0	0	0
Power of:	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Result:	0	64	0	0	8	0	0	0

Total: $64 + 8 = 72$ (H)

Binary (Decimal Calculation)

H -> 01001000

i -> 01101001

EXAMPLE: Decimal Calculation for i

Binary Data:	0	1	1	0	1	0	0	1
Power of:	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Result:	0	64	32	0	8	0	0	1

Total: $64 + 32 + 8 + 1 = 105$ (i)

Answer is: 72

(Verifiera med ASCII Tabellen)

72

Verify: <https://www.asciitable.com/>

71	47	107	G	G
72	48	110	H	H
73	49	111	I	I
74	50	112	J	J



ASCII-tabellen är helt byggd på binärspråk

Base64 Bonus (Valfritt)



Base64 & Binary (Problem)

Tänk dig att du har rå binär data ... bara 0or och 1or.

Datorer älskar det!

Men när du vill skicka den informationen över
internet (*som i JSON eller en JWT*) behöver du text.. inte
slumpmässiga 'byte'

HTTP, tycker inte om detta!

010011100101



Base64

(Why was it invented?)

“Old protocols like **HTTP**, **SMTP (email)**, and **JSON** were designed to handle **only ASCII text..** printable characters:

A–Z,

a–z,

0–9, and a few symbols.”



Base64 (Why?)

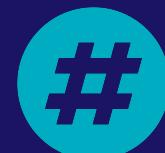
Det är vanligt att skicka text över nätet, vi ser det i form av textdokument, JSON samt requests och responses.

I många fall är data inte enbart text!

Binär data (*som bilder, filer eller krypterad data*) innehåller tecken som inte kan skrivas ut, vilket kan bryta eller korrumpera överföringsprotokoll (som HTTP).

När vi skapar en **JWT** används vår **secret** för att **beräkna en hash (signatur)** av datan.. inte för att själv bli hashad.

Denna signatur blir binär data, som sedan **Base64URL-kodas** för att kunna skickas som text.



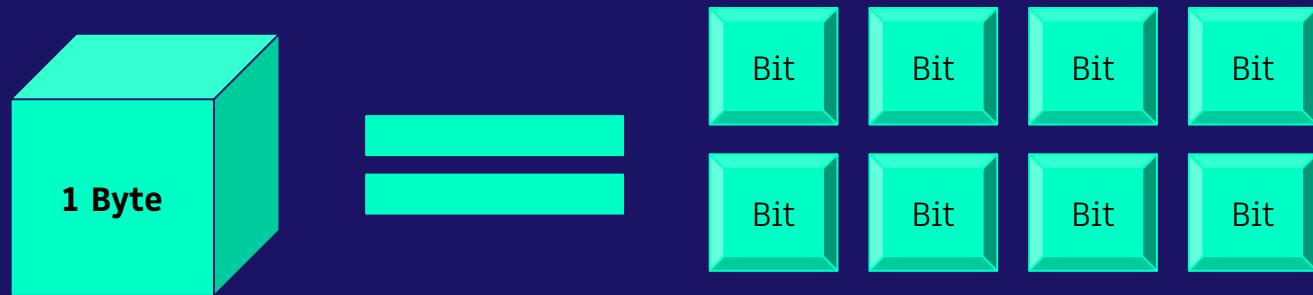
010011100101



SGVs bG8gd29ybGQK

Bytes & Bits

(Index Part #0)



```
private byte binaryData;
```

8 bits
(chunks)

FUN FACT – BONUS

(Byte, Bits & Booleans)

```
// --- A byte in Java = 8 bits ---
// Each bit represents a binary digit (0 or 1)
byte myByte = 0b01010101; // binary literal for 85
System.out.println("Byte value: " + myByte);
System.out.println("Each byte = 8 bits");

// --- A boolean logically = 1 bit (true/false) ---
boolean flagTrue = true;
boolean flagFalse = false;

// But physically, Java stores it as a full byte (8 bits)
// Because CPU cannot address less than one byte in memory.
System.out.println("\nBooleans (logical 1 bit, stored as 1 byte):");
System.out.println("flagTrue = " + flagTrue);
System.out.println("flagFalse = " + flagFalse);
```

FUN FACT #2 – BONUS

(Manipulate Bits within Bytes)

```
// --- BitSet: packs booleans efficiently using 1 bit each ---
java.util.BitSet bits = new java.util.BitSet ();
bits.set(0, true);    // 1st bit = true
bits.set(1, false);   // 2nd bit = false
bits.set(2, true);    // 3rd bit = true

System.out.println ("\nBitSet (stores booleans as bits, not bytes):" );
System.out.println ("Bit 0 = " + bits.get(0));
System.out.println ("Bit 1 = " + bits.get(1));
System.out.println ("Bit 2 = " + bits.get(2));

// BitSet compresses booleans into real bits inside a byte array internally
System.out.println ("\nBitSet internal representation uses ~"
    + bits.size() + " bits total (padded to byte boundaries)." );
}

}
```

Internally (stored in one long = 64 bits): 00000000 00000000 00000000 00000000 00000000 00000000 00000000 10000101

Base64 – Alphabet

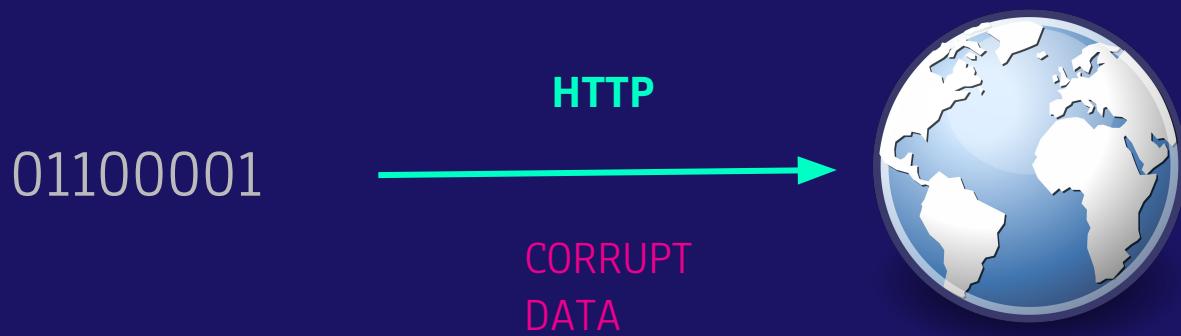
(Index Part #1)

Base64 alfabetet

Index: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/-

26 + 26 + 10 + 2

Base64 - Binary over HTTP = NO (Index Part #2)



Base64: Over HTTP = YES

(Index Part #3)

01100001



Base64

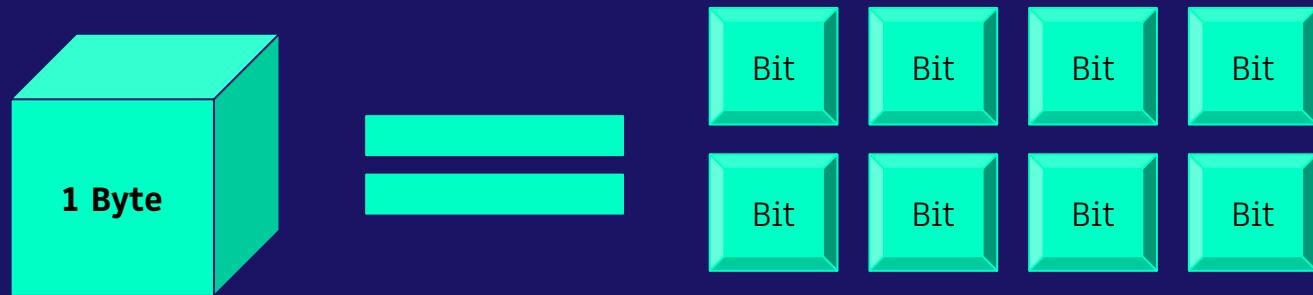
HTTP

SUCCESS



Bytes & Bits

(Binary Data)



Binär data kommer oftast idag i
form av 1 Byte (8 bits).
Men är inte begränsad till 1 byte

8 bits
(chunks)

Bytes & Bits

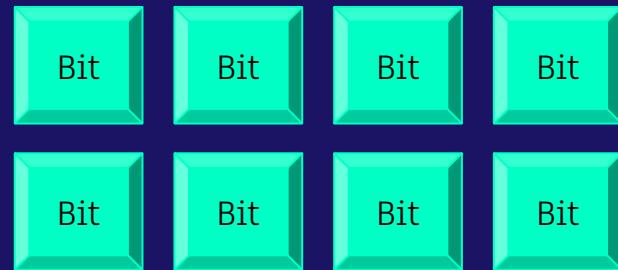
(Binary, not always 8 bits)

Base64: 6 bits per symbol

Base32: 5 bits per symbol

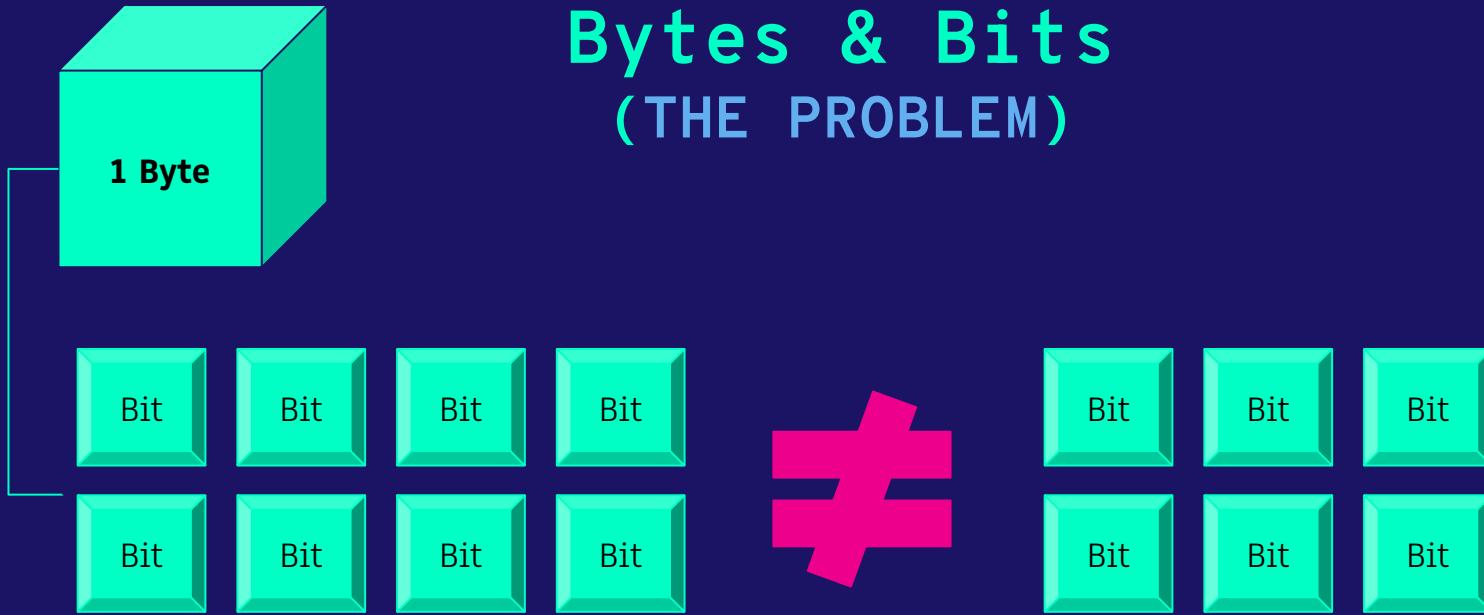
TCP flags: 1 bit each

Color channels: sometimes 10 or
12 bits per pixel component



8 bits
(chunks)

Bytes & Bits (THE PROBLEM)



Binary Data

6 bits

(Base64)

Base64 – Binary Conversion (Index Part #3)

Problem: Binärd data som: 01100001, kan INTE bryta ner till base64..
(6-bit chunks krävs för att tolka base64)

Alltså godtyckligt blir istället 011000 01
Detta blir ju såklart ett problem då vi har 01 kvar..



```
01011101010010
10001010110101
01010010101111
01010010010100
01101010010101
```

Base64 – Binary Example (Index Part #4)

Paste text or drop text file

```
hi
```

Character encoding (optional)

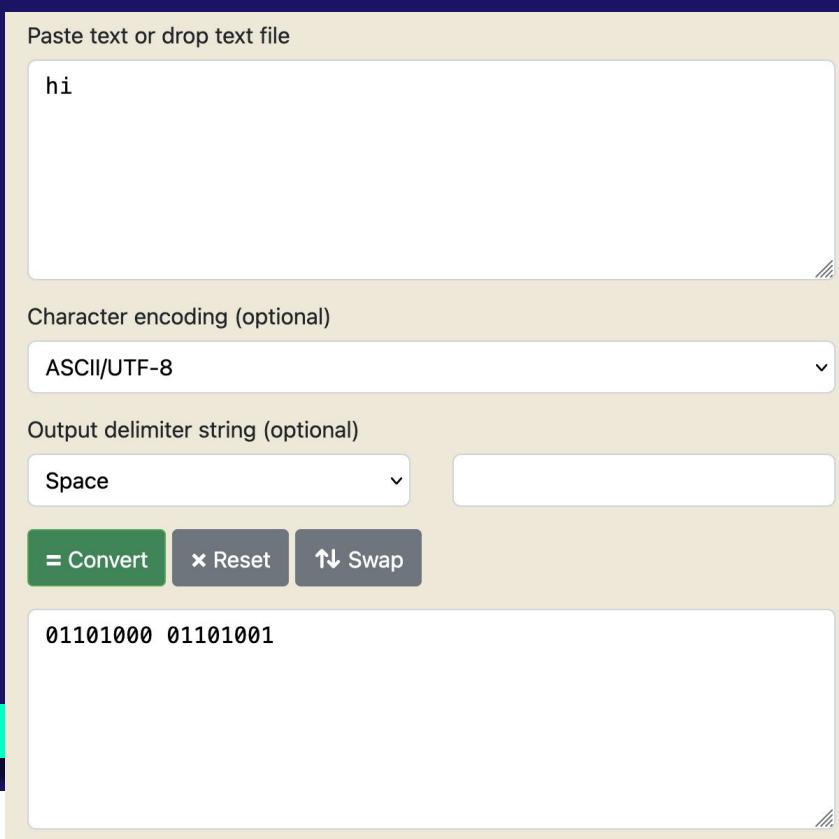
ASCII/UTF-8

Output delimiter string (optional)

Space

= Convert x Reset ⚡ Swap

```
01101000 01101001
```



Result: 01101000 01101001
Base 64: 011010 000110 1001??

Website used:

<https://www.rapidtables.com/convert/number/ascii-to-binary.html>

Base64 – Calculation Table (Index Part #5)

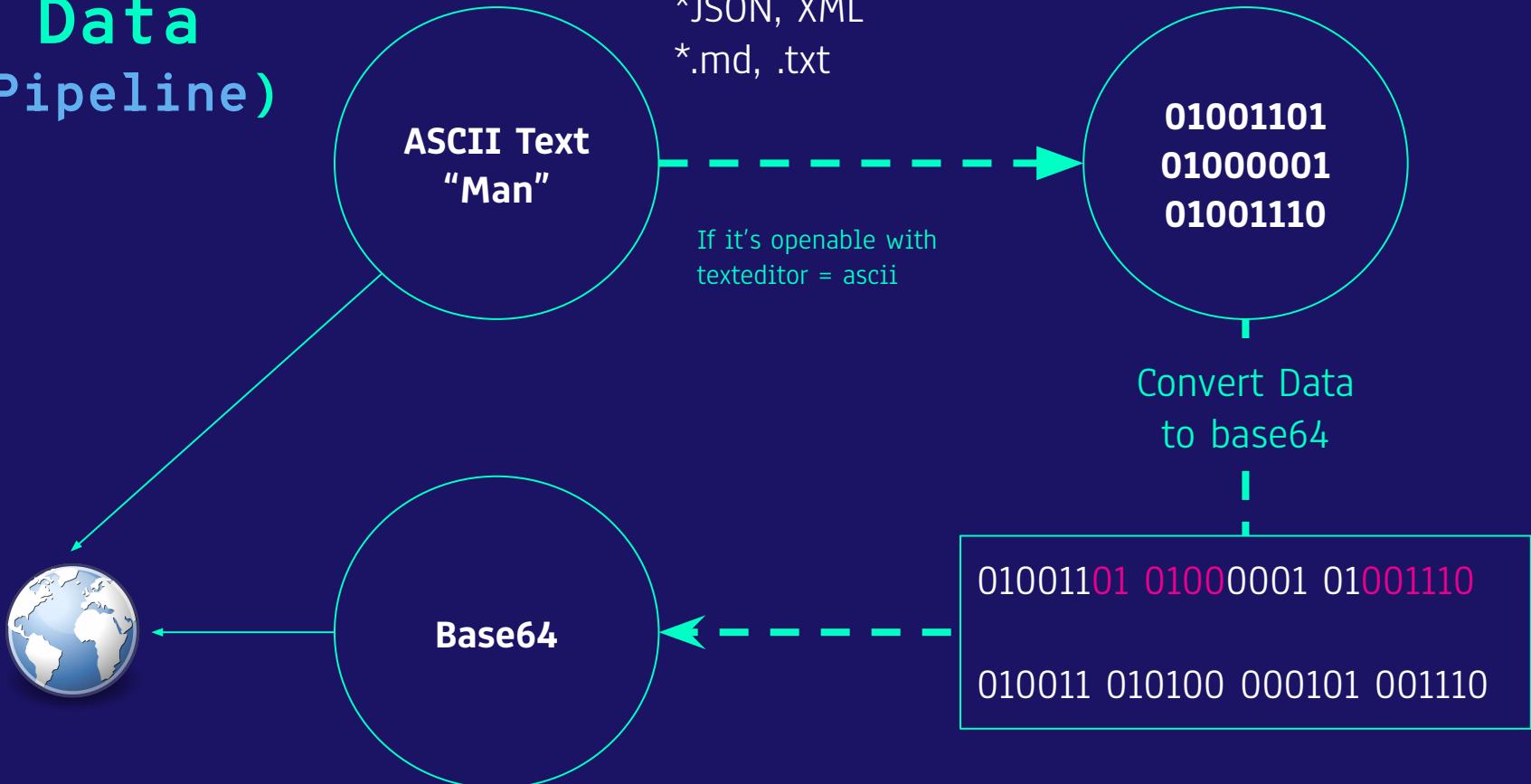
$$(x * 2^5) + (y * 2^4) + (z * 2^3) + (a * 2^2) + (b * 2^1) + (c * 2^0)$$

$$011010 = 0 + 16 + 8 + 0 + 2 + 0 = \mathbf{26}$$

$$000110 = 0 + 0 + 0 + 4 + 2 + 0 = \mathbf{6}$$

$$1001?? = 32 + 0 + 0 + 4 + ? + ? = ? \mathbf{36}$$

Data (Pipeline)



Base64 (Problem)

01001000 - H
01101001 - i

010010 000110 100100 = when there's not
enough space, append with 00

Convert to binary:

<https://www.rapidtables.com/convert/number/ascii-to-binary.html>

010010 = S
000110 = G
100100 = k
Padding: =

Index	Binary	Char									
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/

Convert to binary:

<https://www.rapidtables.com/convert/number/ascii-to-binary.html>

Final Padding (=):

Base64 encoding requires padding with = if the total number of bits isn't a multiple of 6. In this case, we only had 16 bits (two 6-bit groups and one 4-bit group), so we need to pad with one =.

<https://www.base64decode.org/>

Try writing SGk=

Convert to binary:

<https://www.rapidtables.com/convert/number/ascii-to-binary.html>

Base64 (Summarized)

Data Type: Binary data (*Java's 'byte' type is perfect for Binary Data*)

6-bit Chunks: In Binary, each character (byte) consist of 8-bits

Alphabet: A-Z, a-z, 0-9, +, /

Uppercase letters A-Z (26 characters)

Lowercase letters a-z (26 characters)

Digits 0-9 (10 characters)

Two additional characters: + and / (for the final 2 values in the 6-bit range).

Totalt 64 symboler - det är här namnet base64 kommer ifrån

THANKS !

Do you have any questions?
kristoffer.johansson@sti.se

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.