

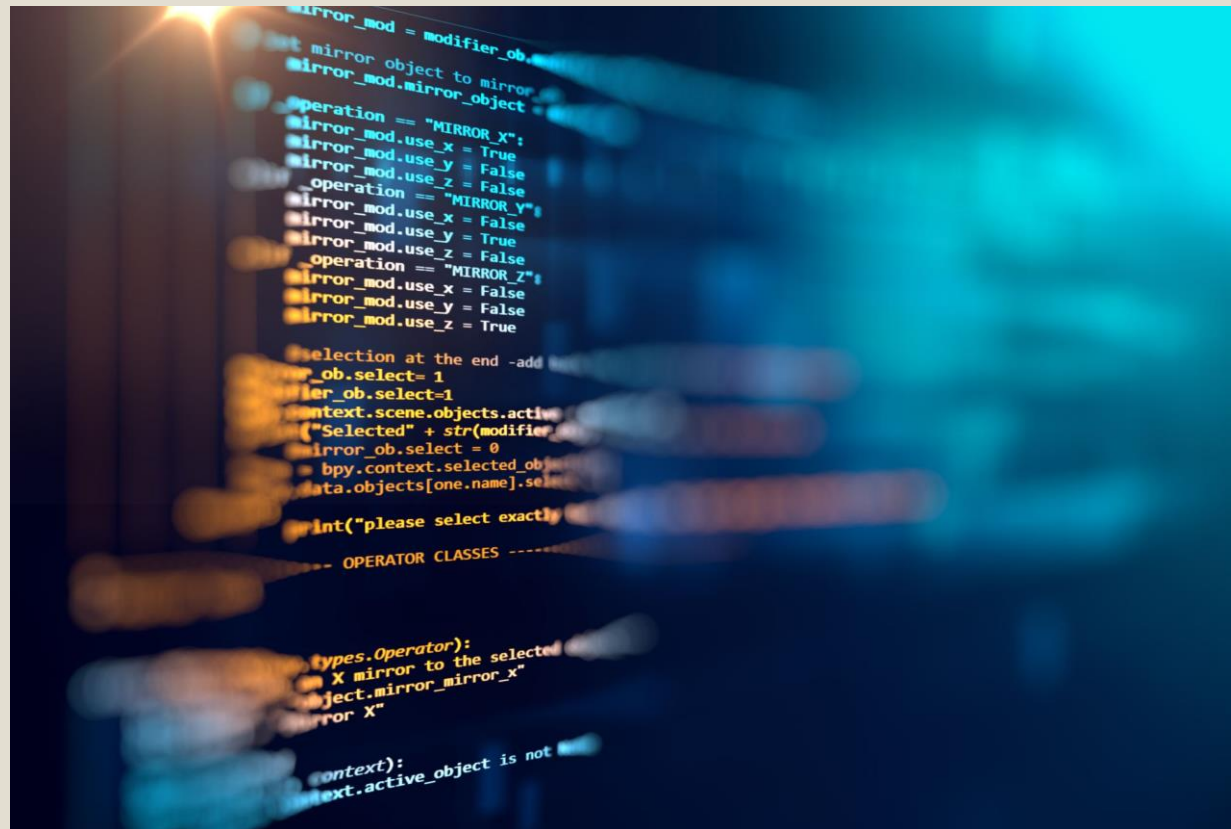


PROGRAMMERING FÖR UX-PRODUKTION 01

Intro + Om web

Välkommen till kursen

Programmering för UX produktion



Lärarna

- Mikael Engström mikael.engstrom@nackademin
- Marcus Johansson marcus.johansson@nackademin.se

Micke Engström - CV

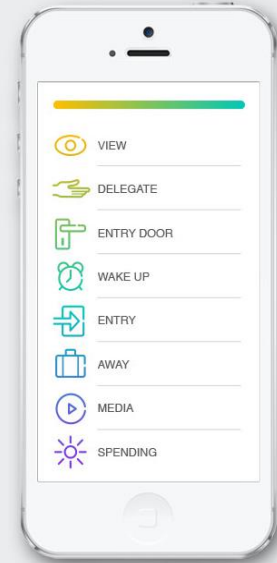
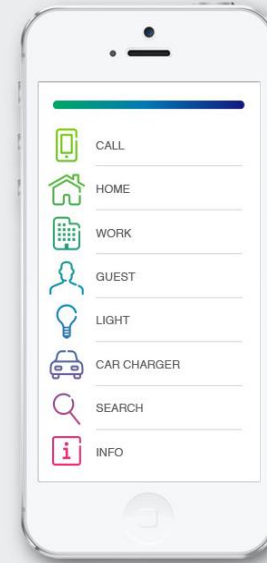
- Netsite – Webb och kurser
- TV Söder – TV, kurser och webb
- Primat – Webb och mobilappar
- Adimo – Webb, mobilappar och CMS
- How Solutions
 - Svenska Spel
- Kronofogdemyndigheten
- MYH
- Kurser
 - Foghorn
 - Zocom



Kunskaper

- ASP.NET/C#
- Frontend
 - HTML
 - CSS
 - Javascript
- Internet of things (IOT)
 - C++
- Wordpress
 - PHP

Exempel



MARINSAJTEN.SE

START NYHETER TV HAMNAR

Logga in

Konstfack

01 About Konstfack

02 Programmes

03 Apply

04 Contact

05 News

ESSINGEN

Sajten för oss på Essingeöarna

Essingeöarna är förmodligen världens bästa plats att bo.

Essingeöarna

Idag 17° Fredag 21°

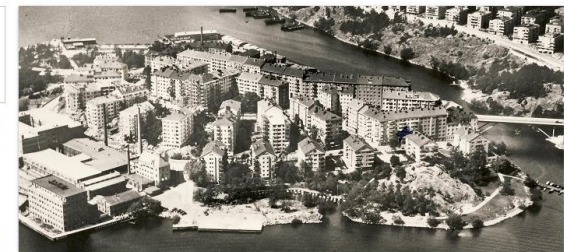
Vind: 3 m/s Vind: 5 m/s
Data från yr.no, MI, NRK Detaljerad prognos

Essingetorget

1	Frihaanen	4 min
1	Frihaanen	14 min
96	Hornsberg via Fridhemsplan	16 min
1	Frihaanen	24 min

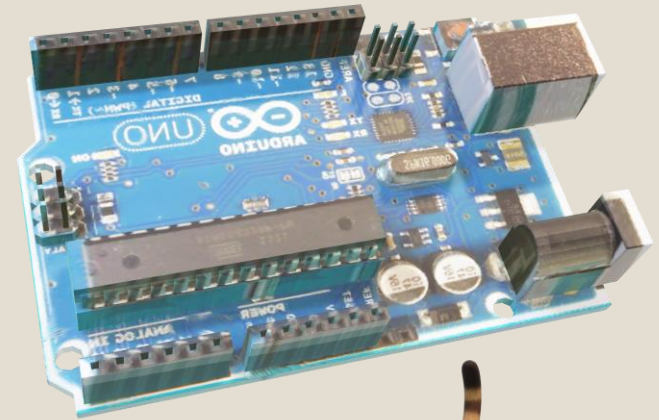
Tvårbanan

22	Sickla	1 min
26	Östermalm	2 min



Pizzeria IL 08-6190740

Micke Engström - Privat



Eleverna

- Namn och kort presentation
- Byggt Webbssidor?
 - HTML, CSS och Javascript
- Programmerat?
 - Programspråk?
- Förväntningar på kursen?



Allmänna frågor

- Kan hantera GIT?
 - Github Pages
- Utvecklingsverktyg
 - Visual Studio Code?



Kursen

- Programmering för UX produktion
- 20 YHP
 - 4 veckor: 14/9 - 7/10
 - 2,5 dagar lektion/veckan
 - Övrig tid: uppgifter/övningar och egenstudier
- Förkunskapskrav inför kurs: Nej

Kursens innehåll

- Kursens syfte:
 - Syftet med kursen är att ge de studerande en övergripande förståelse för de centrala märkningsspråk (HTML och CSS) och programmeringsspråk (Javascript) som styr utseende och interaktion på webben.
- Kursens mål:
 - Målet med kursen är att de studerande kan ställa relevanta krav på frontend-utvecklare och skapa UX-design som är genomförbar att utveckla i kod. De studerande ska få grundläggande kunskaper om programmering i HTML och CSS för att få förståelse för utvecklarens arbete i syfte att kunna samverka med utvecklare.
- De studerande ska ha färdigheter i att:
 - Påvisa förståelse hur man kan skapa en webbplats uppdelat i HTML och CSS-filer.
 - Påvisa kunskap hur man kan använda Javascript
- De studerande ska ha kompetens för att:
 - Välja lösningar som fungerar i ett brett perspektiv för många användare och tekniska enheter samtidigt som det skapar en bättre upplevelse och gynnar projektmål.

Examination

- Examinationsform
 - Inlämningsuppgifter och gruppuppgift.
- Betygskriterier
 - Betyget Icke godkänd (IG)
 - sätts om studenten inte uppnått samtliga kriterier för Godkänd. För Godkänd (G): krävs att studenten får godkänt på följande kriterier:
 - För betyget Godkänd (G)
 - ska den studerande:
 - Ha förståelse för hur innehåll och utseende på en webbplats byggs upp i separata filer.
 - Ha förståelse för hur programmering kan användas för att manipulera innehåll och utseende i syfte att skapa interaktion och animationer som förbättrar användbarhet och tillgänglighet.
 - Skapa en webbplats uppdelat i HTML och CSS-filer.
 - Skapa dynamiska prototyper med stöd för interaktivitet som ger ökad förståelse för intressenter i ett projekt samt ett tydligt underlag för utveckling och produktion.
 - Välja lösningar som fungerar i ett brett perspektiv för många användare och tekniska enheter samtidigt som det skapar en bättre upplevelse och gynnar projektmål.
 - För Väl Godkänd (VG):
 - krävs att studenten får godkänt på samtliga G – respektive följande VG-kriterier:
 - För betyget Väl godkänd (VG) ska den studerande:
 - Med ett nyskapande tillvägagångssätt argumentera för hur programmering kan användas för att manipulera innehåll och utseende i syfte att skapa interaktion och animationer som förbättrar användbarhet och tillgänglighet

Kursmoment

- Vecka 1
 - Hur fungerar:
 - Webbläsare
 - Webserver
 - HTTP
 - En sajts beståndsdelar
 - HTML, CSS, Javascript
 - HTML - Hur fungerar det
 - HTML-dokumentets struktur
 - Vanligaste HTML-elementen
 - CSS - hur styr man utseendet på sidan
 - Layout-egenskaper
 - Gruppuppgift
 - Bygg en första webbsida

Kursmoment

- Vecka 2
 - Mer HTML
 - Formulär
 - Semantisk HTML
 - Knappar
 - Mer CSS
 - Bootstrap
 - Knappdesign
 - Gruppuppgift
 - Bygga en webbsida

Kursmoment

- Vecka 3
 - Javascript
 - Syfte och funktion
 - DOM
 - CSS
 - Animationer
 - Gruppuppgift
 - Bygga en interaktiv webbsida
 - Inlämningsuppgift presenteras
- Vecka 4
 - Repetitioner
 - Arbete med inlämningsuppgift

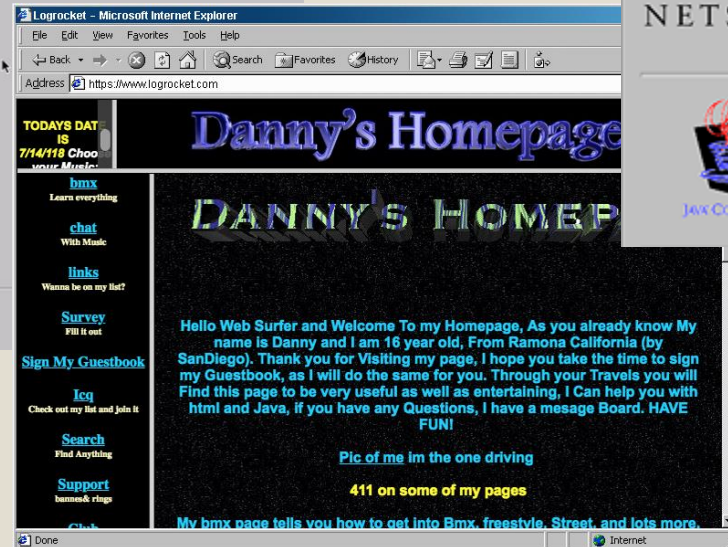
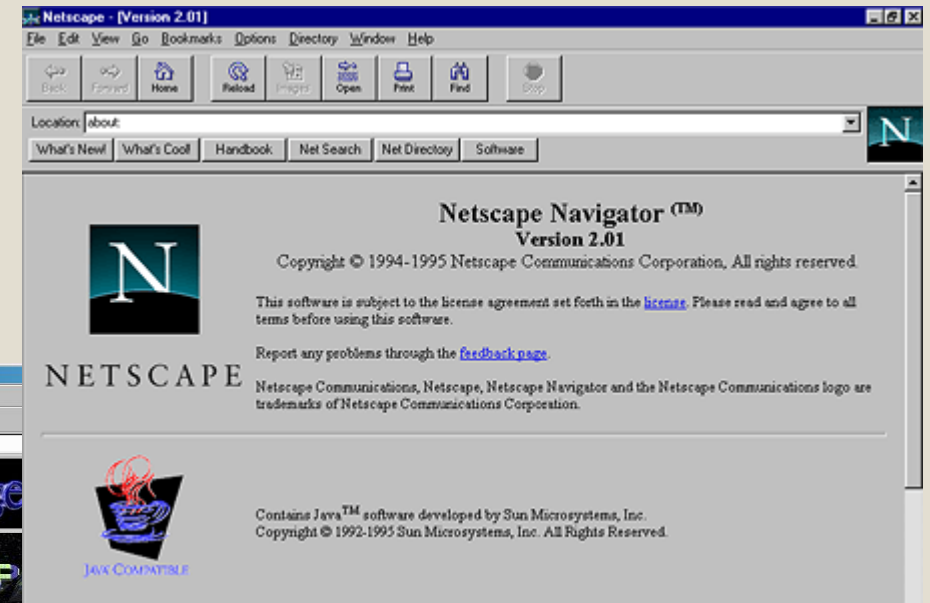
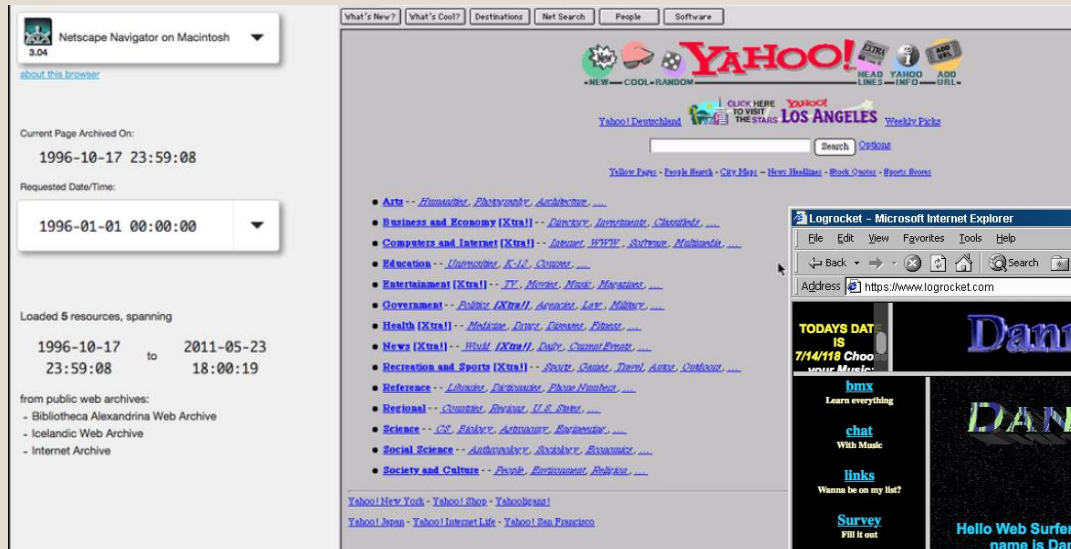
Idag

- Alla presenterar sig
- Kursen presenteras
- Hur fungerar:
 - Webbbläsare
 - Webserver
 - HTTP
- En sajts beståndsdelar
 - HTML, CSS, Javascript
- 12 Regler + grupparbete

Webbläsaren

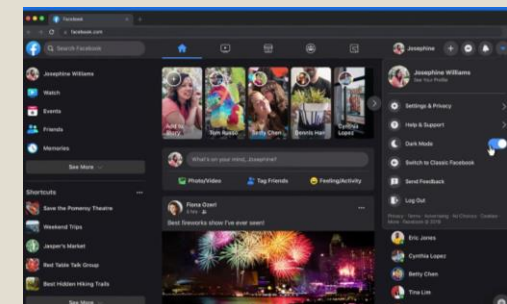
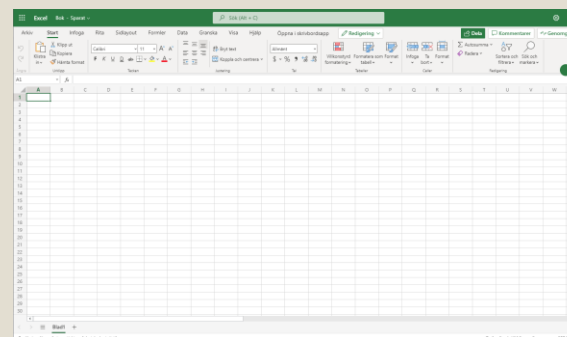
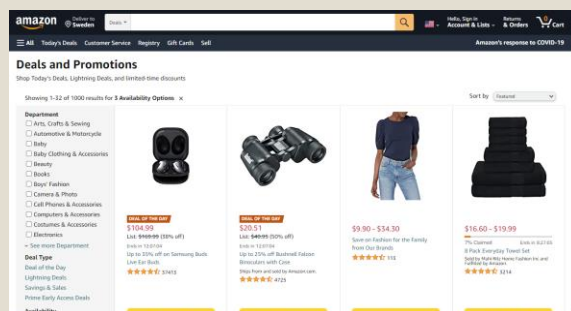
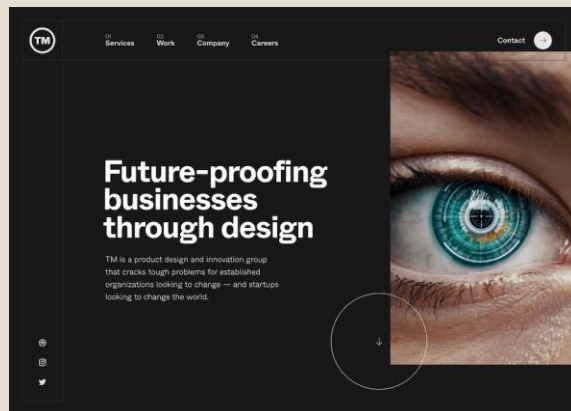
- Programmet som visar webbsidor
- Förr i tiden kunde webbläsaren enbart visa text och länkar
- Sedan tillkom bilder
- Idag kan webbläsaren visa, kort sagt, alla typer av innehåll, köra komplexa program och hantera stora mängder data
- Webbläsaren, eller webbrowsern, är idag det program vi använder mest på en mobiltelefon och dator.

Gamla webbsidor



The Internet As It Was In 1996 - 90's Websites - YouTube

Webbsidor idag



Vad webbläsare kan göra idag

- Web GL
 - [Infinitown | A WebGL Experiment by Little Workshop](#)
 - [WipEout \(PSX\) – Model Viewer \(phoboslab.org\)](#)
- Applikationer
 - <https://www.office.com/>

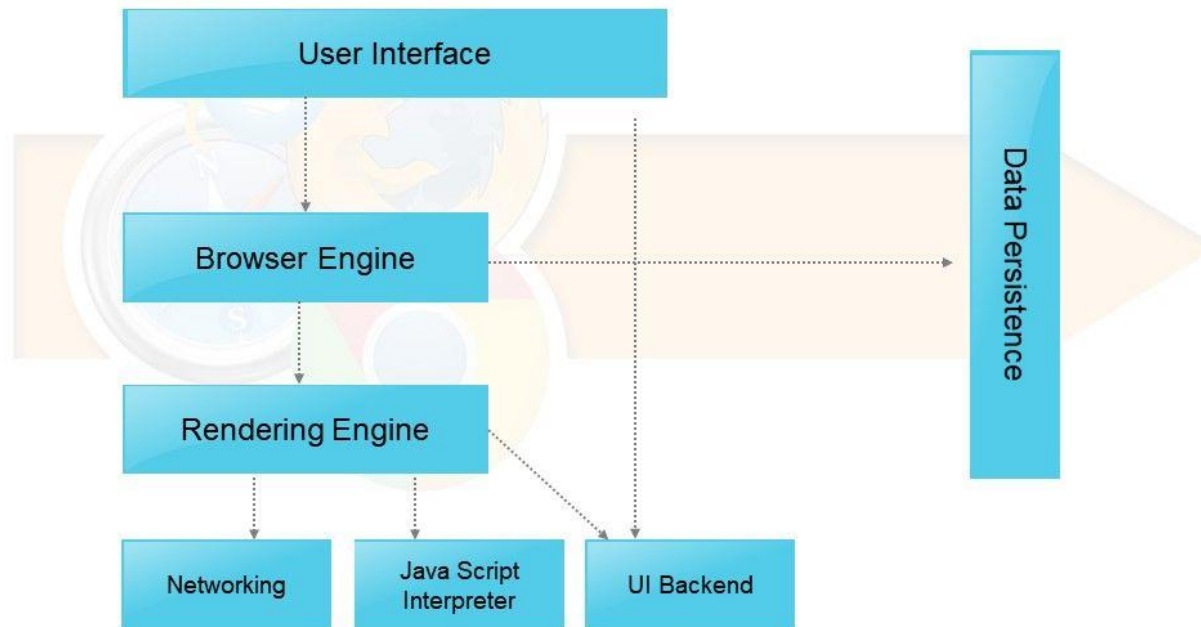
Varför bättre nu?

- Ökat kunnande
- Kraftfullare datorer och mobiltelefoner
- Snabbare uppkoppling
- Ökad efterfråga
- Men framför allt: Mycket större möjligheter i webbläsaren
 - Som UX-designer är det bra att veta vad som är möjligt
 - Dialog med utvecklare
 - Veta lite själv om tekniken
 - Förstå hur mycket jobb det är

Lite teknik om webbläsaren

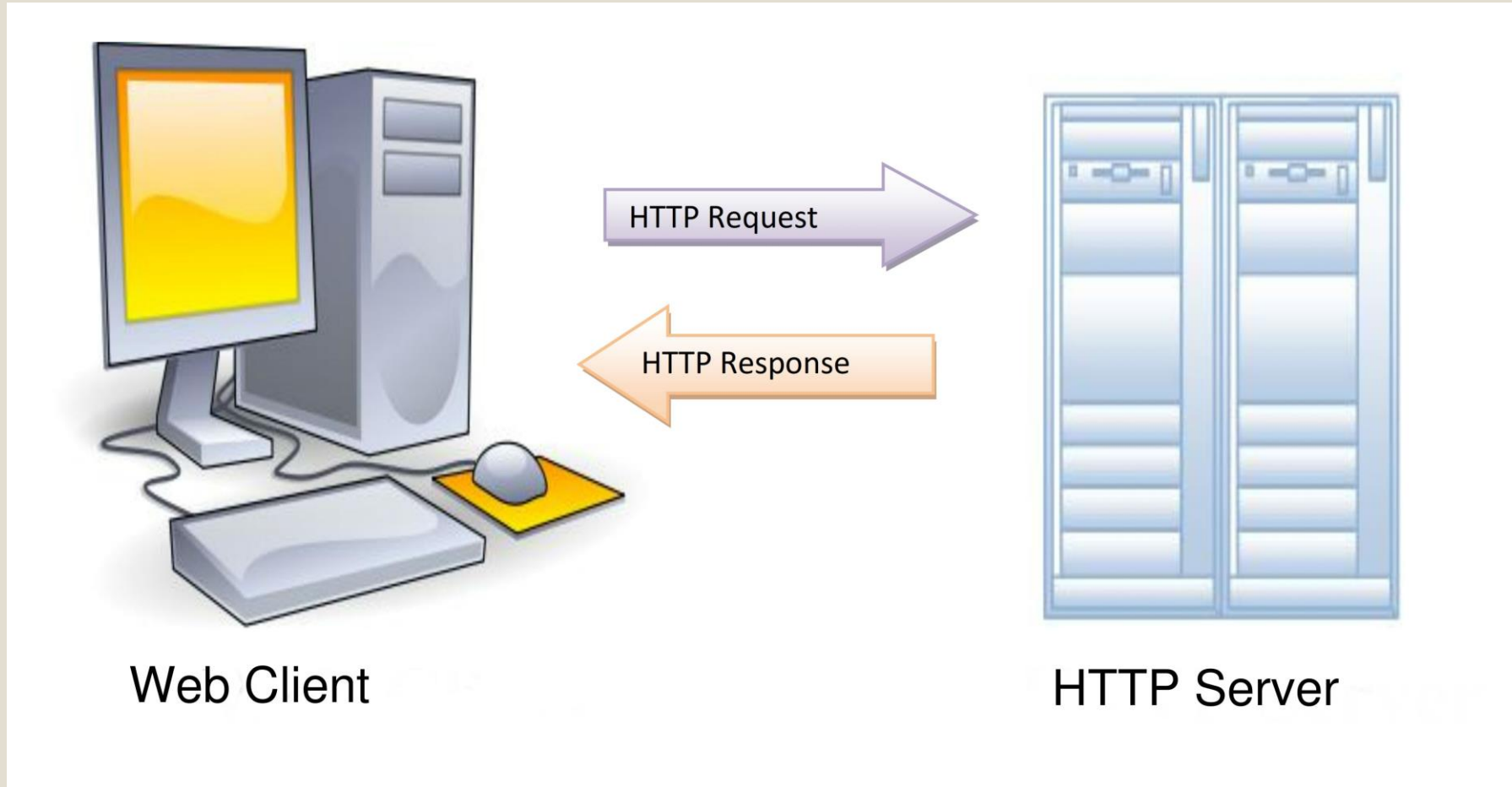
Block Diagram Showing Web Browser Architecture

This slide is 100% editable. Adapt it to your need and capture your audience's attention.



*UI – User Interface

Lite om hur webbsidan transporteras



Webbserver

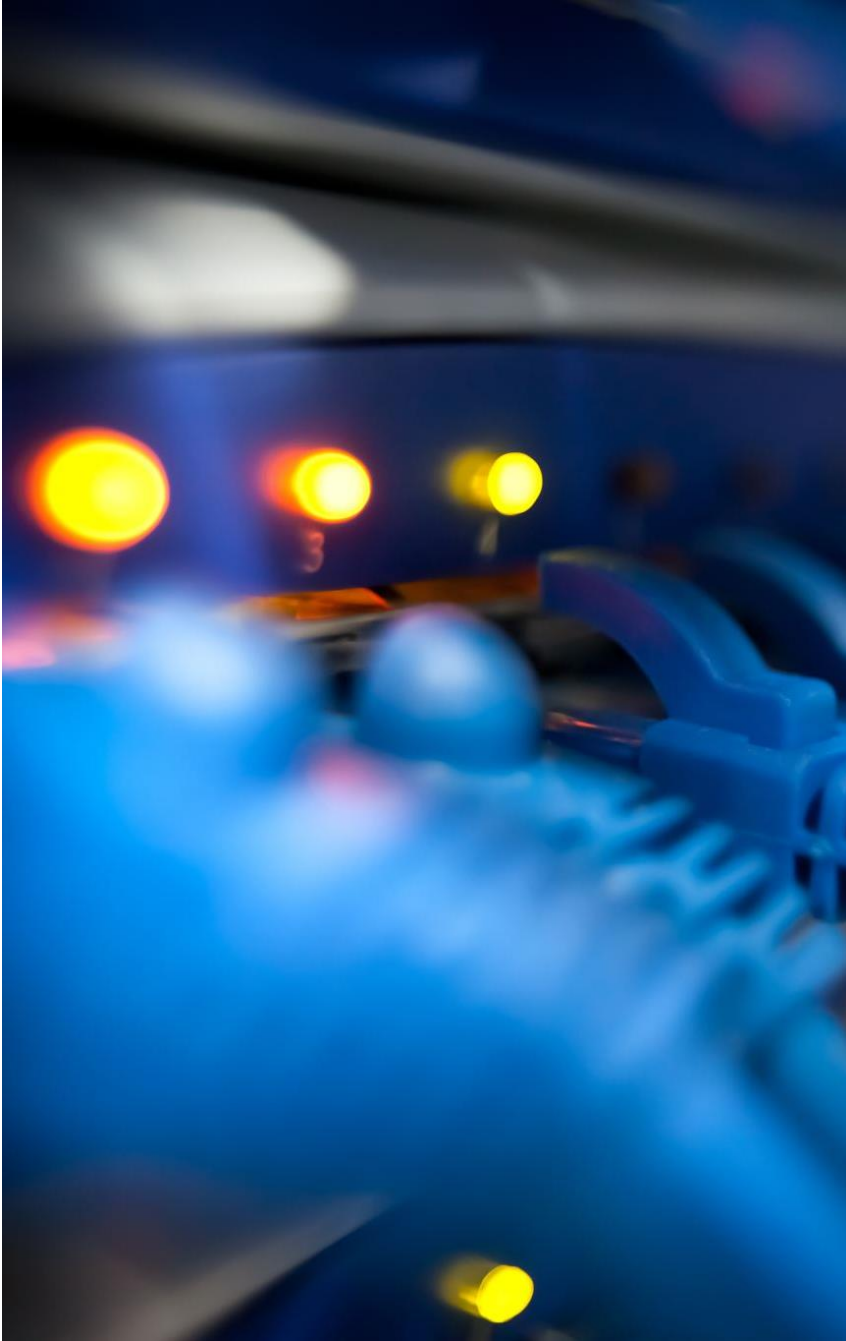
- En webserver är en dator som är ansluten till Internet.
 - Vår egen dator kan agera webserver
 - En webbläsare ansluter till en annan dator, och ber om innehåll.
 - Det kan vara
 - Webbssidor HTML
 - CSS Filer som talar om hur webbsidan ska se ut
 - Javascript Filer som talar om hur webbsidan ska fungera
 - Övriga filer
 - Bilder JPG, PNG, GIF, SVG
 - Filer Alla olika typer av filer (word, pdf osv)
 - Media Video och ljud

Webbserver

- Det finns olika webserver-plattformar

- | | | |
|-------------------------------|-----------|--------------------|
| ◦ Internet information Server | Microsoft | exempelvis ASP.NET |
| ◦ Nginx | NGINX | |
| ◦ Apache | Apache | |

- Men, i sin enklaste form är det bara en enkel filsstruktur på din lokala dator, som du sedan klickar på.



HTTP – Hypertext transfer protocol

- Hypertext Transfer Protocol (HTTP) är det kommunikationsprotokoll som används för att överföra webbsidor på informationsnätverket WWW, World Wide Web på Internet. Det ursprungliga syftet med HTTP var att tillhandahålla en metod för att överföra HTML-sidor från webbservrar till webbklienter. Standard från 1996.
- Hypertext = text som innehåller länkar till andra texter. Myntat 1965

HTTP

- HTTP definierar åtta kommandon som en klient kan skicka till en resurs på en HTTP-server:
 - **GET** – Ber servern att skicka den utpekade filen (eller resultatet av en programkörning, databasförfrågan eller motsvarande) till klienten. Detta är i särklass det mest använda HTTP-kommandot.
 - **HEAD** – Ber servern att skicka information om den utpekade resursen, men utan att skicka själva innehållet i filen.
 - **POST** – Sänder någon form av information till servern, utöver själva förfrågan, vanligen motsvarande en webblankett. Detta används dels då förfrågan ändrar information på servern, såsom då man gör en bokning, dels då innehållet väntas vara större än vad som är lämpligt att hantera som en webbadress (eller annars olämpligt, såsom lösenord).
 - **PUT** – Om tillåtelse ges, laddar upp en utpekad fil från klienten till servern för lagring.
 - **DELETE** – Raderar den utpekade filen. Detta kommando används sällan och många webbservrar har inget stöd för det.
 - **TRACE** – Ber servern att skicka tillbaka klientförfrågan precis i det skick som den anlände till servern. Detta kommando kan användas för att kontrollera om någon tredje part mellan klient och server har gjort några ändringar i förfrågan.
 - **OPTIONS** – Returnerar en lista över de HTTP-kommandon som servern stöder.
 - **CONNECT** – Används med proxy-servrar som kan fungera som SSL-tunnlar.

HTTP

Ett http-anrop:

- Först skickar webbläsaren ett request:
- **GET** / HTTP/1.1 (GET /**dokument/start.html** HTTP/1.1)
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134
Accept-Language: en-US,en;q=0.8,sv-SE;q=0.5,sv;q=0.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Host: 192.168.0.194
Connection: Keep-Alive

Servern svarar med att skicka en header

"HTTP/1.1 200 **OK**\r\n"

"Content-Type: text/html\r\n"

"Connection: close\r\n" "\r\n"

HTTP Exempel

Klienten (t ex webbläsaren):

GET /index.html HTTP/1.1

Host: www.example.com

Serverns svar:

HTTP/1.1 200 OK Date: Mon, 23 May 2005 22:38:34 GMT

Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)

Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT

Etag: "3f80f-1b6-3e1cb03b"

Accept-Ranges: bytes

Content-Length: 438

Connection: close

Content-Type: text/html; charset=UTF-8

[blankrad]

<html><head></head><body>...

HTTP/HTTPS

- HTTPS är http-protokollet, men krypterat.
- Fram till för ett antal år sedan var all överföring av webbsidor i läsbart textformat.
- Idag ska all HTTP-trafik/leverans av webbinnehåll vara krypterat.
- Länk till krypterat innehåll ser ut Såhär:
 - <https://www.google.se/>
- Google indexerar inte längre http-webbsidor.

Summering Webbbläsare, server och HTTP

- En webserver kan vara vilken dator som helst, som är ansluten till Internet
- Att skicka webbsidor sker idag via textbaserade protokoll från 90-talet
- Webbsidor skickas som “enkla” HTML-textfiler, som sedan visas i webbläsaren
- Hur kan den här enkla strukturen leverera de webbsidor vi ser idag?
 - Webbläsarens inbyggda funktionalitet
 - CSS
 - Javascript
 - Mediafiler: bilder, video osv

Webbplatsens tre delar

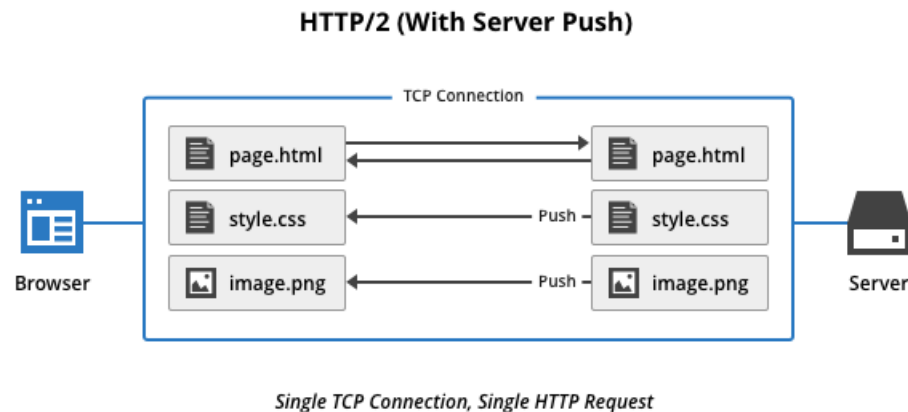


Webbsidans tre delar

- **HTML** (Hyper Text Markup Language) beskriver en sidas innehåll (content) och struktur.
"De här elementen ska finnas på sidan, i den här ordningen"
- **CSS** beskriver hur taggar/element ska presenteras.
"Det här elementet ska ha röd textfärg och vara understruket."
- **JavaScript** hanterar användarens interaktion med sidan.
"När jag klickar på knappen ska det här hända."
- **Separation of concerns** innebär att man inte vill blanda HTML, CSS och JavaScript utan ha dem i separata filer. Filerna slutar normalt med .html, .css och .js



Webbsidans beståndsdelar



- Den fasta punkten i en webbsida är HTML-dokumentet.
- Den laddas in i webbläsaren
- Därefter går webbläsaren igenom HTML-dokumentet för att se om det behövs fler filer.
- De filer som behövs hämtar webbläsaren i tur och ordning.
- När alla externa filer är laddade kan webbsidan visas som den är tänkt, med all funktionalitet.

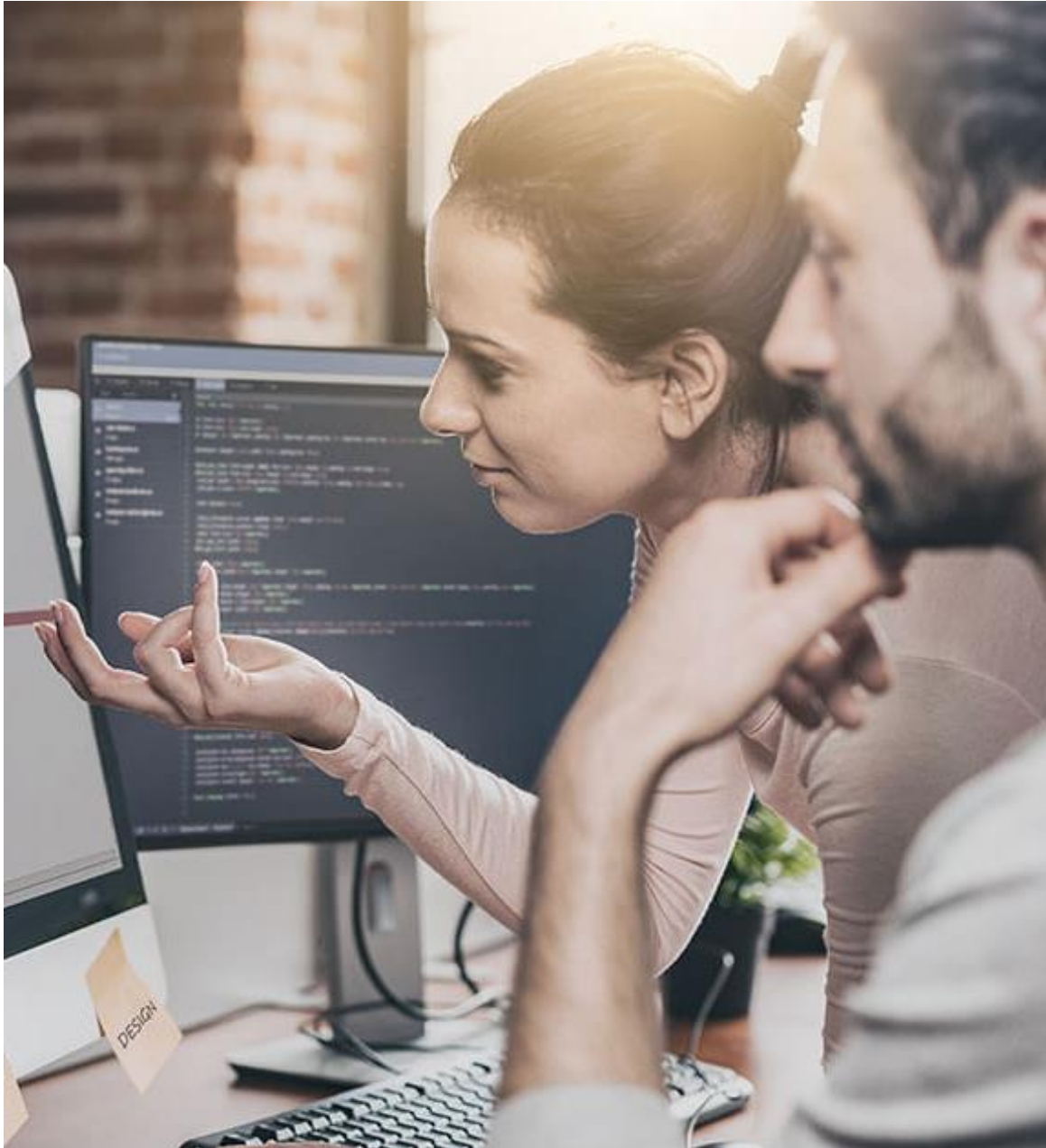
Programvara

- Programmet vi använder för att bygga webbsidor heter Visual Studio Code
- Laddas ner gratis här: [Visual Studio Code - Code Editing. Redefined](#)
- Läxa till nästa gång: Installera Visual Studio Code
- Läxa 2: Installera tillägget "Live Server" av Ritwik Dey



Demo – The Lamp App (spoiler alert)

- Vi ska titta på ett komplett enkelt exempel på en HTML-sida, som länkar in bild, CSS och Javascript.
- Syftet är INTE att ni ska förstå koden redan nu, utan bara få ett smakprov på vad vi kommer prata om under kursen.
- Syftet är också att se hur en filstruktur kan se ut.
- Kod: `index.html`



12 Regler

- I den här kursen kommer ni prova på att vara utvecklare.
- Syftet är:
 - att ni kan ställa relevanta krav på frontend-utvecklare
 - skapa UX-design som är genomförbar att utveckla i kod.
 - få grundläggande kunskaper om programmering i HTML och CSS
 - få förståelse för utvecklarens arbete i syfte att kunna samverka med utvecklare
- Separat PDF: 12 Regler (Lära sig programmera)

Nästa gång

- HTML - Hur fungerar det
 - HTML-dokumentets struktur
 - Vanligaste HTML-elementen
 - Enkel egen webbsida utan design
- Fram tills dess:
 - Titta igenom länkarna på nästa sida
 - Om du vill tjuvstarta, så finns även länkar för det på nästa sida.
 - Genomgången på torsdag kommer dock anta att ni inte tittat på de sidorna.

Länkar

- [How does web browsers work?. A browser is a software application... | by Monica Raghuwanshi | Medium](#)
- [What is a Web Server and How Does it Work? \(techtarget.com\)](#)
- [HTTP and HTML \(video\) | Khan Academy](#) (Kolla videon)
- Tjuvstart:
 - HTML: [HTML Tutorial \(w3schools.com\)](#)
 - CSS: [CSS Tutorial \(w3schools.com\)](#)
 - Javascript: [JavaScript Tutorial \(w3schools.com\)](#)