

# Första webbservern

- Internet består av en massa sammankopplade datorer
- Varje dator har en IP-adress (v4), t.ex. `197.251.9.120` (kallas ibland för *hostname*)
  - Din dators "egna" IP-adress är vanligen `127.0.0.1` (aka *localhost*) eller `0.0.0.0`
- För att komma åt en dator anger vi även en *port*, t.ex. `197.251.9.120:80`
  - Portar är olika ingångar till samma plats, t.ex. HTTP använder port 80
  - Portar finns inom spannet 0 till 65535
  - De första 1024 portarna är reseverade
- *Domäner* gör det enklare att komma ihåg dessa platser
  - Såsom `mau.se` eller `google.com`

# Första webbservern

- En webbserver sägs *lyssna* på en port efter förfrågningar, för att kunna svara på dessa
- Vanligen anger vi port 8000, 8080 eller 8888, för att det är enkelt att komma ihåg
- **API:** [`Deno.serve\(handler: ServeHandler<Deno.NetAddr>\)`](#)
- Vi anropar `serve` med en funktion (*handler*) som tar emot en [`request`](#) och returnerar en [`response`](#)
  - Request och response är klasser som representerar en förfrågan eller ett svar
- Standard för Deno är att webbservern lyssnar på `0.0.0.0:8000`, alternativt `localhost:8000`
- Vi behöver ange `--allow-net` för att vårt program ska få lov att fungera

# Första webbservern

```
function handler(request) {  
    const response = new Response("Hello Web!");  
    return response;  
}  
  
Deno.serve(handler);
```

*Det är ungefär samma princip som för `addEventListener`*

# Första webbservern

```
Deno.serve((request) => {  
    const response = new Response("Hello Web!");  
    return response;  
});
```

*Det är ungefär samma princip som för `addEventListener`*

# Svara med HTML

```
function handler(request) {  
  const html = "<h1>Hello Web!</h1>";  
  const options = {  
    status: 200,  
    statusText: "OK",  
    headers: { "Content-Type": "text/html" }  
  };  
  const response = new Response(html, options);  
  return response;  
}  
  
Deno.serve(handler);
```