

// introduktion til

# **PROGRAMMERING/KODNING & IT-TANKEGANG (5/5)**



**Claus Brabrand**

( [brabrand@itu.dk](mailto:brabrand@itu.dk) )

( <http://www.itu.dk/people/brabrand/> )

Full Professor, Ph.D.  
Head of <Center for Computing Education Research>  
Department of Computer Science  
 IT University of Copenhagen, Denmark

<http://studwww.itu.dk/~brabrand/ITU-PC/>

<https://tinyurl.com/itupc>

# KURSUSPLAN

**Fredag (28/02):** 09:30 – 15:30

**HTML & CSS**

**Fredag (07/03):** 09:30 – 15:30

**HTML Forms & Kodning...**

**Fredag (14/03):** 09:30 – 15:30

**Mere kodning...**

**Fredag (21/03):** 09:30 – 15:30

**Endnu mere kodning...**

**Fredag (28/03):** 09:30 – 15:30

**Databaser & Kommunikation & Web Services**

# KURSUSPLAN

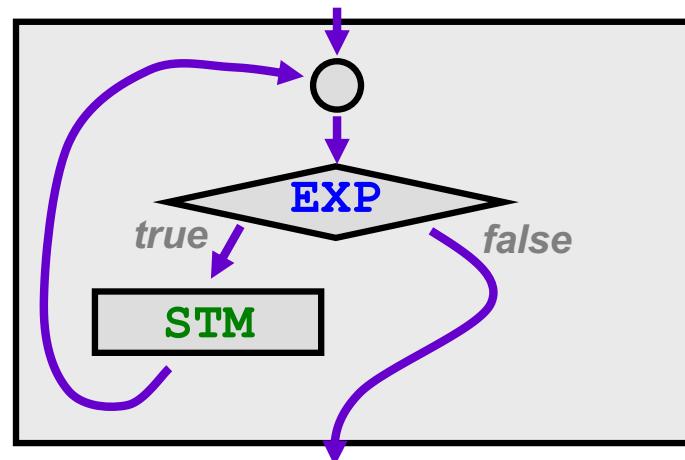
# **Resumé**

# Iteration: 'while'

Syntaks:

```
while ( EXP ) STM
```

Semantik:



Eksempel:

```
var i = 0;
while ( i < 10 ) {
    document.write("<li>" + i + "</li>");
    i = i + 1;
}
```

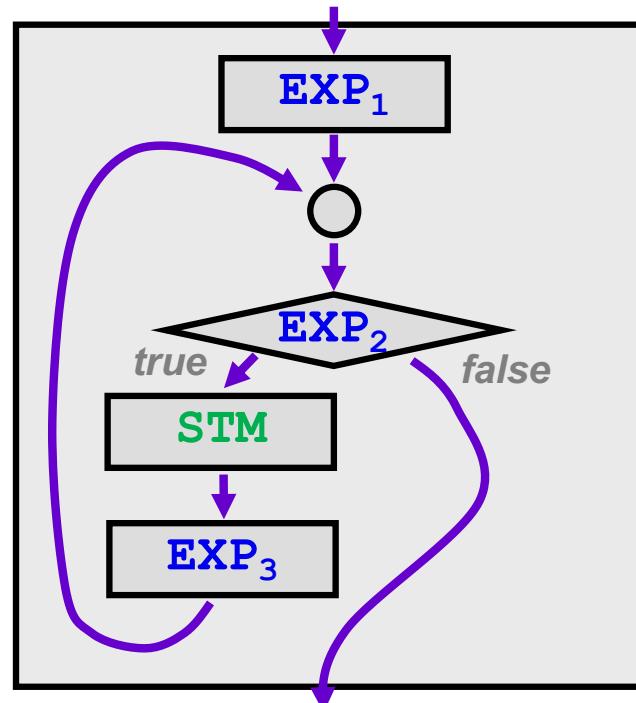
- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

# Iteration: 'for'

Syntaks:

```
for ( EXP1 ; EXP2 ; EXP3 ) STM
```

Semantik:



- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

Eksempel:

```
for ( var i = 0 ; i < 10 ; i = i + 1 ) {  
    document.write("<li>" + i + "</li>");  
}
```

# Gennemløb af Strenge

Operationer på tekst-strenge:

<code>s.length</code>	// længden af <code>s</code>
<code>s.charAt(0)</code>	// første tegn i <code>s</code>
<code>s.charAt(s.length-1)</code>	// sidste tegn i <code>s</code>

```
<html>
  <body>
    <code>
      <script> 012
        ↓↓
        var s = "Strenge gennemløbes og udskrives ét tegn ad gangen.";
        for ( var index = 0 ; index < s.length ; index++ ) {
          var tegn = s.charAt(index); //find index'te tegn i strengen
          document.write(tegn);     // udskriv tegn'et
        }
      </script>
    </code>
  </body>
</html>
```

<https://www.w3schools.com/code/tryit.asp?filename=FVCJS34I272Y>

Strenge gennemløbes og udskrives ét tegn ad gangen.

# Grammatik for JavaScript

<b>EXP</b> expression (udtryk)	→ KONSTANT → VAR → EXP + EXP	7 x x + 1
<b>STM</b> statement (sætning)	→ NAVN = EXP ; → var NAVN = EXP ; → if ( EXP ) STM → if ( EXP ) STM else STM → while ( EXP ) STM → for ( EXP ; EXP ; EXP ) STM → { STMS } → NAVN ( ARGS ) ;	x=x+1 ;

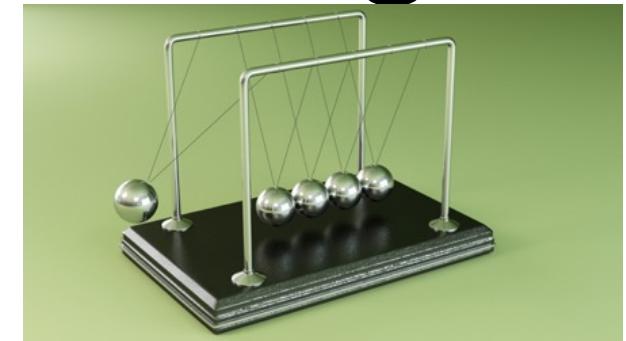
# Grammatik for Udtryk (Exp)

EXP	→	KON	7
expression (udtryk)	→	VAR	x
	→	EXP + EXP	x + 1
	→	EXP * EXP	2 * 2
	→	EXP - EXP	y - 1
	→	EXP / EXP	1 / 3
	→	EXP == EXP	x == 0
	→	EXP != EXP	x != 1
	→	EXP > EXP	x > 0
	→	EXP < EXP	y < 10
	→	EXP >= EXP	x >= y
	→	EXP <= EXP	0 <= x
	→	EXP && EXP	a && b
	→	EXP    EXP	a    b
	→	EXP % EXP	x % 2
	→	! EXP	! true
	→	( EXP )	( x>0 )

# Fundamentale Begrænsninger

"Evighedsmaskinen":

Som aldrig ville kunne laves



Den strider mod fundamentale naturlove

(Termodynamikkens anden lov)

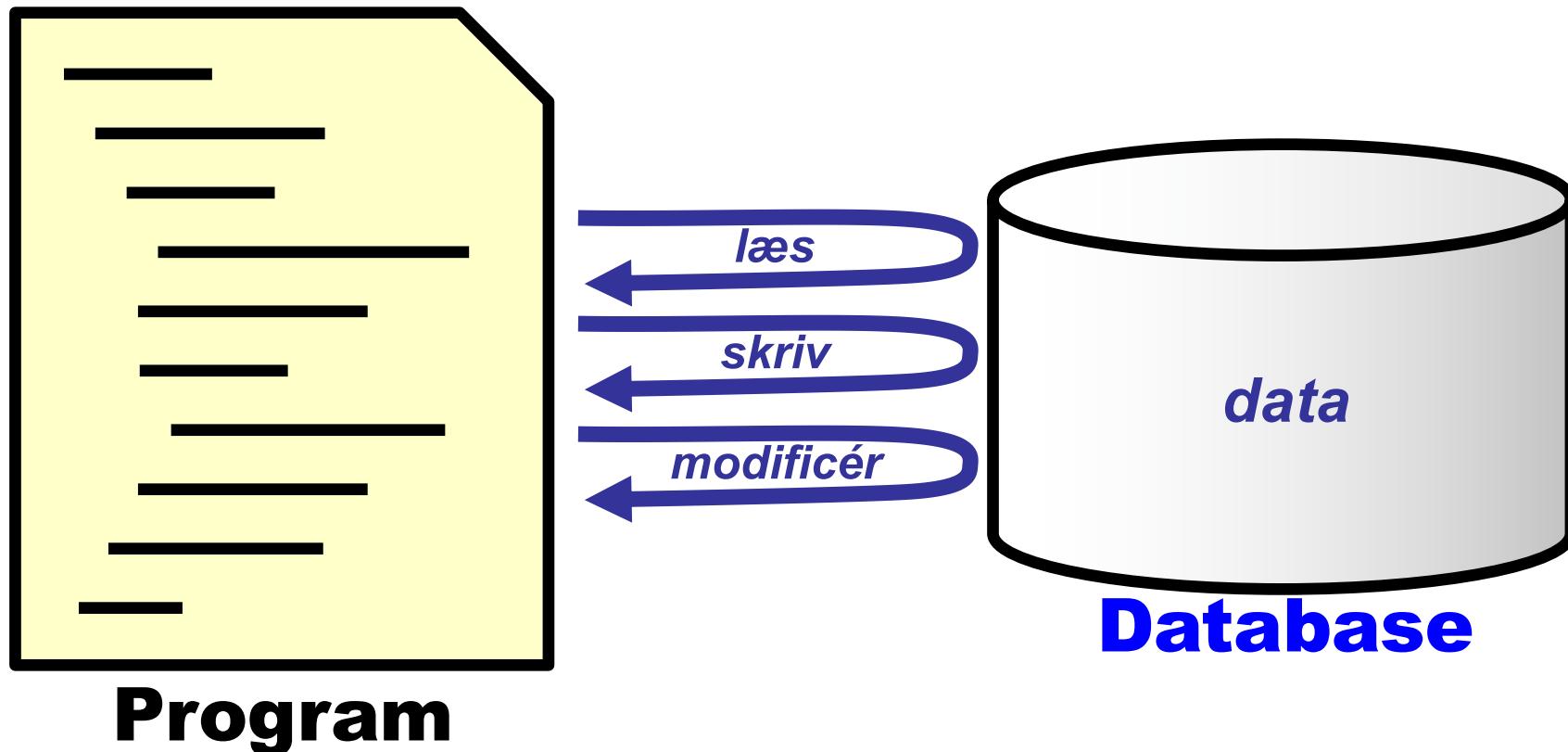
Tilsvarende findes der nogle **fundamentale begrænsninger** på **hvad en computer kan** (og nogensinde ville kunne) **løse / beregne!**

- 1) *Umulige problemer*
- 2) *Upraktiske problemer*
- 3) *Upræcise problemer*

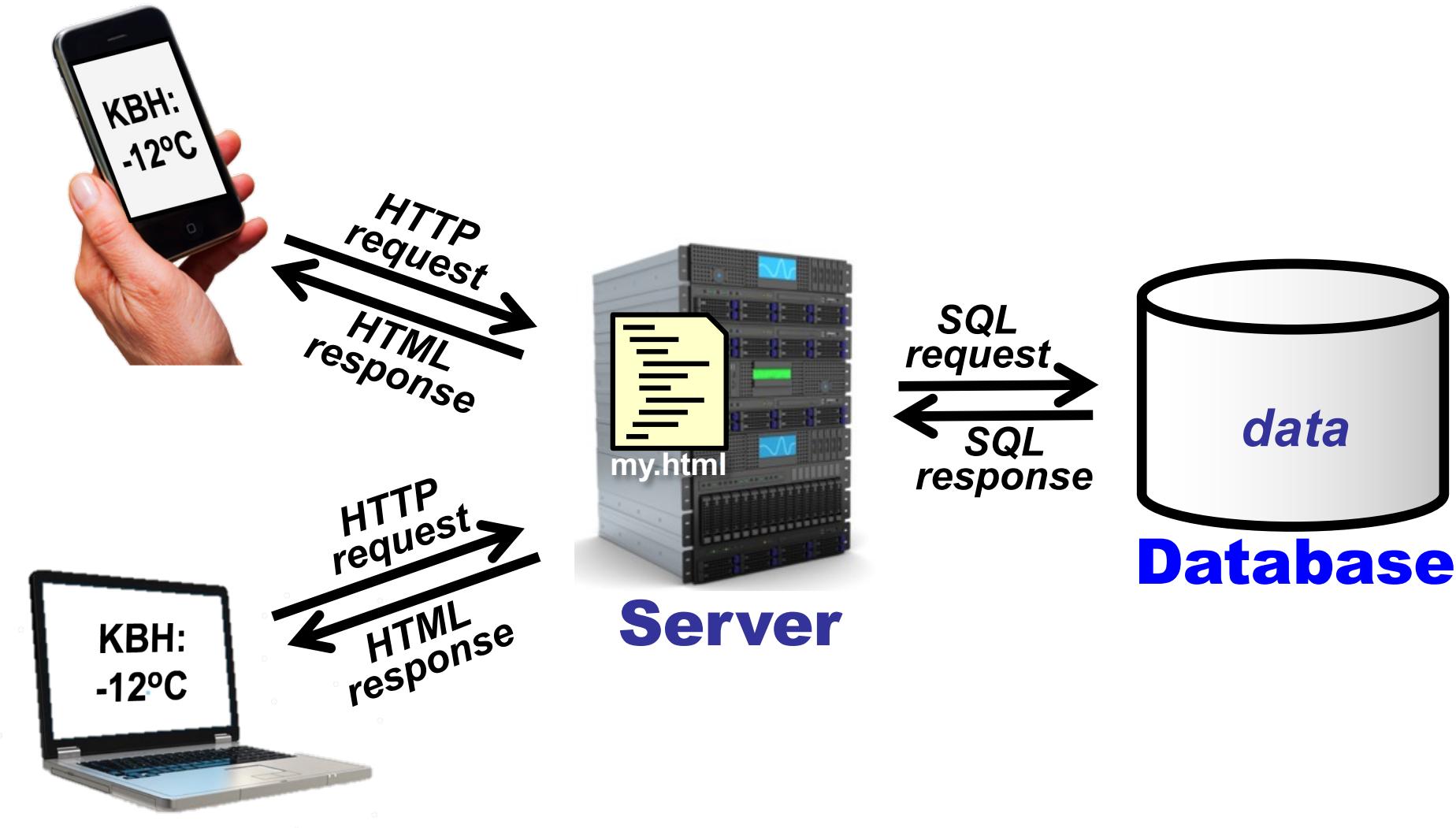
## Fundamental Limitations

# **Databaser (SQL)**

# Program ↔ Database



# Web Arkitektur (med Database)



## SQL er et *database-programmeringssprog*:

Opfundet i 1970'erne og populært siden 1980'erne

Mest almindelige måde at tilgå relationelle databaser på

Oprindeligt "SEQUEL" (Structured English Query Language)

Nu: "SQL" (Structured Query Language)

### Eksempel:

```
SELECT first_name FROM users WHERE last_name = 'Hansen';
```

## MySQL (gratis implementation af SQL):

***robust, hurtig, gratis***

( [www.mysql.com](http://www.mysql.com) )

...og brugt rigtigt meget i softwareindustrien

# Tabel (Entitet)

Customers: ← Tabellens navn

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany
7	Blondel père et fils	Frédérique Citeaux	24, place Kléber Citeaux	Strasbourg	67000	France
8	Bólido Comidas preparadas	Martín Sommer	C/ Araquil, 67	Madrid	28023	Spain
9	Bon app'	Laurence Lebihans	12, rue des Bouchers	Marseille	13008	France
10	Bottom-Dollar Marketse	Elizabeth Lincoln	23 Tsawassen Blvd.	Tsawassen	T2F 8M4	Canada

række  
(record):

kolonne:

Navn: 'CustomerName'  
& Type: tekststreng

felt:  
inneholder data!

# w3school Online SQL (Try-It)

SQL Statement:

```
SELECT * FROM Customers;
```

## SQL Editor:

Her kan man skrive  
et "database query"

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Kør database  
query'et ovenfor

Result:

Resultat af  
database query

Number of Records: 91

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Your Database:

Tablename	Records
Customers	91
Categories	8
Employees	10
OrderDetails	518
Orders	196
Products	77
Shippers	3
Suppliers	29

Restore Database

Oversigt  
over  
tabeller i  
databasen

[https://www.w3schools.com/sql/trysql.asp?filename=trysql\\_select\\_all](https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all)



# SELECT FROM



Få databasen til at udskrive **alle (ti) ansatte**:

EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
1	Davolio	Nancy	1968-12-08	EmpID1.pic	Education includes a BA in psychology from Colorado State University. She also completed (The Art of the Cold Call). Nancy is a member of 'Toastmasters International'.
2	Fuller	Andrew	1952-02-19	EmpID2.pic	Andrew received his BTS commercial and a Ph.D. in international marketing from the University of Dallas. He is fluent in French and Italian and reads German. He joined the company as a sales representative, was promoted to sales manager and was then named vice president of sales. Andrew is a member of the Sales Management Roundtable, the Seattle Chamber of Commerce, and the Pacific Rim Importers Association.
3	Leverling	Janet	1963-08-30	EmpID3.pic	Janet has a BS degree in chemistry from Boston College). She has also completed a certificate program in food retailing management. Janet was hired as a sales associate and was promoted to sales representative.
4	Peacock	Margaret	1958-09-19	EmpID4.pic	Margaret holds a BA in English literature from Concordia College and an MA from the American Institute of Culinary Arts. She was temporarily assigned to the London office before returning to her permanent post in Seattle.

( HINT: Tabellens navn hedder: "Employees" )



# OPGAVE



# SELECT-FROM

```
SELECT CustomerName, City FROM Customers;
```

CustomerName	City
Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim
Blondel père et fils	Strasbourg
Bólido Comidas preparadas	Madrid
Bon app'	Marseille
Bottom-Dollar Marketse	Tsawassen
B's Beverages	London
Cactus Comidas para llevar	Buenos Aires

[https://www.w3schools.com/sql/trysql.asp?filename=trysql\\_select\\_columns](https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_columns)



# SELECT FROM



1) Udskriv kunders  
*residensland*:

Country
Germany
Mexico
Mexico
UK
Sweden
Germany
France
Spain
France
Canada
UK
Argentina

1

2) Udskriv ansattes  
*fornavn & fødselsdag*:

FirstName	BirthDate
Nancy	1968-12-08
Andrew	1952-02-19
Janet	1963-08-30
Margaret	1958-09-19
Steven	1955-03-04
Michael	1963-07-02
Robert	1960-05-29
Laura	1958-01-09
Anne	1969-07-02
Adam	1928-09-19

2



# OPGAVE



# SELECT-FROM-WHERE

```
SELECT CustomerName, City FROM Customers WHERE Country='Brazil';
```

CustomerName	City
Comércio Mineiro	São Paulo
Familia Arquibaldo	São Paulo
Gourmet Lanchonetes	Campinas
Hanari Carnes	Rio de Janeiro
Que Delícia	Rio de Janeiro
Queen Cozinha	São Paulo
Ricardo Adocicados	Rio de Janeiro
Tradição Hipermercados	São Paulo
Wellington Importadora	Resende

I SQL skrives  
lig med: '='  
(altså ikke '==')

[https://www.w3schools.com/sql/trysql.asp?filename=trysql\\_select\\_columns](https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_columns)

# SELECT FROM WHERE

Udskriv *produktnavn*, *enhed* og *pris* på alle produkter fra *produktkategori* nummer 2:

ProductName	Unit	Price
Aniseed Syrup	12 - 550 ml bottles	10
Chef Anton's Cajun Seasoning	48 - 6 oz jars	22
Chef Anton's Gumbo Mix	36 boxes	21.35
Grandma's Boysenberry Spread	12 - 8 oz jars	25
Northwoods Cranberry Sauce	12 - 12 oz jars	40
Genen Shouyu	24 - 250 ml bottles	15.5
Gula Malacca	20 - 2 kg bags	19.45
Sirop d'éralbe	24 - 500 ml bottles	28.5
Vegie-spread	15 - 625 g jars	43.9
Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21.05
Louisiana Hot Spiced Okra	24 - 8 oz jars	17
Original Frankfurter grüne Soße	12 boxes	13

( HINT: Kig i tabellen: "Products" )

## OPGAVE

# SELECT-FROM-WHERE

Ligesom i **JavaScript**, kan man i SQL benytte **AND** og/eller **OR** til at kombinere (neste) udtryk:

```
SELECT ProductName, Unit, Price FROM Products WHERE CategoryID=2 AND Price<20 ;
```

ProductName	Unit	Price
Aniseed Syrup	12 - 550 ml bottles	10
Genen Shouyu	24 - 250 ml bottles	15.5
Gula Malacca	20 - 2 kg bags	19.45
Louisiana Hot Spiced Okra	24 - 8 oz jars	17
Original Frankfurter grüne Soße	12 boxes	13

[https://www.w3schools.com/sql/trysql.asp?filename=trysql\\_select\\_columns](https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_columns)

# SELECT FROM WHERE

Udskriv **leverandørnavn** og **by** på alle leverandører fra Tyskland **eller** Frankrig:

SupplierName	City
Heli Süßwaren GmbH & Co. KG	Berlin
Plutzer Lebensmittelgroßmärkte AG	Frankfurt
Nord-Ost-Fisch Handelsgesellschaft mbH	Cuxhaven
Aux joyeux ecclésiastiques	Paris
Escargots Nouveaux	Montceau
Gai pâturage	Annecy

( HINT: Kig i tabellen: "Suppliers" )



## OPGAVE



# Kombinationer af Udtryk

Nestede udtryk med **AND** og/eller **OR**:

```
SELECT ProductName, Unit, Price FROM Products WHERE  
CategoryID=2 AND Price<20 ;
```

```
SELECT * FROM Customers WHERE  
Country='Germany' AND (City='Berlin' OR City='München') ;
```

...og **NOT**:

```
SELECT * FROM Products WHERE (NOT (Price<50));
```

```
SELECT * FROM Customers WHERE  
( NOT (Country='Germany') ) AND ( NOT (Country='USA') ) ;
```

(Som altid: hellere for mange paranteser end for få!)

# Sortering

Man kan **sortere** resultaterne fra databasen:

```
SELECT CustomerName, Country, City FROM Customers  
ORDER BY Country, City ;
```

CustomerName	Country	City
Cactus Comidas para llevar	Argentina	Buenos Aires
Océano Atlántico Ltda.	Argentina	Buenos Aires
Rancho grande	Argentina	Buenos Aires
Ernst Handel	Austria	Graz
Piccolo und mehr	Austria	Salzburg
Maison Dewey	Belgium	Bruxelles
Suprêmes délices	Belgium	Charleroi
Gourmet Lanchonetes	Brazil	Campinas

# **SELECT med det hele**

Alle produkter fra *kategori enten 2 eller 3, der ikke koster mindre end 25; sorteret efter pris:*

<b>ProductName</b>	<b>Price</b>
Grandma's Boysenberry Spread	25
Sirop d'érable	28.5
Gumbär Gummibärchen	31.23
Northwoods Cranberry Sauce	40
Schoggi Schokolade	43.9
Vegie-spread	43.9
Tarte au sucre	49.3
Sir Rodney's Marmalade	81

## **OPGAVE**

# Indsæt og Modificér Data

## Indsæt data:

```
INSERT INTO Customers  
  (CustomerName, ContactName, Address, City, PostalCode, Country) VALUES  
  ('ITU', 'Claus Brabrand', 'Rued Langgards Vej 7', 'KBH S', '2300', 'Denmark');  
  
SELECT CustomerName, Country, Address FROM Customers WHERE PostalCode=2300;
```

CustomerName	Country	Address
ITU	Denmark	Rued Langgards Vej 7

## Modificér data:

```
UPDATE Customers  
SET CustomerName='IT University of Copenhagen' WHERE CustomerName='ITU';  
  
SELECT CustomerName, Country, Address FROM Customers WHERE PostalCode=2300;
```

CustomerName	Country	Address
IT University of Copenhagen	Denmark	Rued Langgards Vej 7

# Slet Data

## Slet data:

```
DELETE FROM Customers WHERE PostalCode='2300';
```

```
SELECT CustomerName, Country, Address FROM Customers WHERE PostalCode=2300;
```

No Result.

**Join**

Registersammenkøring

# Lov om offentlige registre



**Lov nr. 294 af 8. juni 1978:** DATATILSYNET

**§ 4.** Registre, der føres for en statslig myndighed, *må kun oprettes i henhold til godkendelse* af vedkommende minister efter forhandling med finansministeren. [...]

**Stk 3.** Bestemmelserne [...] *gælder tilsvarende ved samkøring af registre*, der er oparbejdet med henblik på varetagelse af forskellige opgaver.

(Dette gælder dog ikke samkøring, der udelukkende foretages med henblik på uddrag i statistisk eller videnskabeligt øjemed.)

# Kryds-Produkt: '×'

## ■ *Kryds-produkt* (operator på to *mængder*):

- Lad følgende mængder være givet:

- $A = \{ \text{apple}, \text{pear} \}$

- $B = \{ \text{soccer ball}, \text{basketball} \}$

- $A \times B = \{ (\text{apple}, \text{soccer ball}), (\text{apple}, \text{basketball}), (\text{pear}, \text{soccer ball}), (\text{pear}, \text{basketball}) \}$

- dvs: mængden af alle *par*

## ■ Opgave:

- $A \times A = \{ (\text{apple}, \text{apple}), (\text{apple}, \text{pear}), (\text{pear}, \text{apple}), (\text{pear}, \text{pear}) \}$

- $N \times N = \{ (0,0), (0,1), (0,2), \dots, (1,0), (1,1), \dots, (42,87), \dots \}$

- Hvor mange elementer har mængden:  $X \times Y$ , hvis mængden  $X$  har  $x$  elementer og  $Y$  har  $y$  elementer?

# Example

**mailing\_list:**

name	email
Claus	brabrand@itu.dk
Barack	obama@hotmail.com
John	jdoe@notmail.com

**phone\_numbers:**

email	phone
brabrand@itu.dk	7218 5076
obama@hotmail.com	212-555-0000
jdoe@notmail.com	123-456-7890

```
CREATE TABLE mailing_list ( name VARCHAR(20) NOT NULL, email VARCHAR(50) NOT NULL );
INSERT INTO mailing_list (name, email) VALUES ('Claus', 'brabrand@itu.dk');
INSERT INTO mailing_list (name, email) VALUES ('Barack', 'obama@hotmail.com');
INSERT INTO mailing_list (name, email) VALUES ('John', 'jdoe@notmail.com');
SELECT * FROM mailing_list;
```

```
CREATE TABLE phone_numbers ( email VARCHAR(50) NOT NULL, phone VARCHAR(50) NOT NULL );
INSERT INTO phone_numbers (email, phone) VALUES ('brabrand@itu.dk', '72185076');
INSERT INTO phone_numbers (email, phone) VALUES ('obama@hotmail.com', '212-555-0000');
INSERT INTO phone_numbers (email, phone) VALUES ('jdoe@notmail.com', '123-456-7890');
SELECT * FROM phone_numbers;
```

<https://www.mycompiler.io/new/sql>

[https://www.tutorialspoint.com/online\\_sql\\_editor.htm?preset=example3](https://www.tutorialspoint.com/online_sql_editor.htm?preset=example3)

# Join (1/3)

mailing\_list:

name	email
Claus	brabrand@itu.dk
Barack	obama@hotmail.com
John	jdoe@notmail.com

phone\_numbers:

email	phone
brabrand@itu.dk	7218 5076
obama@hotmail.com	212-555-0000
jdoe@notmail.com	123-456-7890

```
SELECT *
FROM mailing_list , phone_numbers ; // NB: Comma for at joine!
```

name	email	email	phone
Claus	brabrand@itu.dk	brabrand@itu.dk	7218 5076
Barack	obama@hotmail.com	brabrand@itu.dk	7218 5076
John	jdoe@notmail.com	brabrand@itu.dk	7218 5076
Claus	brabrand@itu.dk	obama@hotmail.com	212-555-0000
Barack	obama@hotmail.com	obama@hotmail.com	212-555-0000
John	jdoe@notmail.com	obama@hotmail.com	212-555-0000
Claus	brabrand@itu.dk	jdoe@notmail.com	123-456-7890
Barack	obama@hotmail.com	jdoe@notmail.com	123-456-7890
John	jdoe@notmail.com	jdoe@notmail.com	123-456-7890

"join"  
operation:  
alle  
kombinationer

# Join (2/3)

**mailing\_list:**

name	email
Claus	brabrand@itu.dk
Barack	obama@hotmail.com
John	jdoe@notmail.com

**phone\_numbers:**

email	phone
brabrand@itu.dk	7218 5076
obama@hotmail.com	212-555-0000
jdoe@notmail.com	123-456-7890

```
SELECT *
FROM mailing list , phone numbers // NB: Comma for at joine!
WHERE mailing_list.email = phone_numbers.email ;
```

✓	name	email	email	phone		
✓	Claus	brabrand@itu.dk	brabrand@itu.dk	7218 5076		
✗	Barack	obama@hotmail.com	brabrand@itu.dk	7218 5076		
✗	John	jdoe@notmail.com	brabrand@itu.dk	7218 5076		
✗	Claus	brabrand@itu.dk	obama@hotmail.com	212-555-0000		
✓	Barack	obama@hotmail.com	obama@hotmail.com	212-555-0000		
✗	John	jdoe@notmail.com	obama@hotmail.com	212-555-0000		
✗	Claus	brabrand@itu.dk	jdoe@notmail.com	123-456-7890		
✗	Barack	obama@hotmail.com	jdoe@notmail.com	123-456-7890		
✓	John	jdoe@notmail.com	name	email	email	phone
		Claus	brabrand@itu.dk	brabrand@itu.dk	7218 5076	
		Barack	obama@hotmail.com	obama@hotmail.com	212-555-0000	
		John	jdoe@notmail.com	jdoe@notmail.com	123-456-7890	

"join" operation:  
alle kombinationer  
ok, nu har vi rækkerne :-)

# Join (3/3)

**mailing\_list:**

name	email
Claus	brabrand@itu.dk
Barack	obama@hotmail.com
John	jdoe@notmail.com

**phone\_numbers:**

email	phone
brabrand@itu.dk	7218 5076
obama@hotmail.com	212-555-0000
jdoe@notmail.com	123-456-7890

```
SELECT name, mailing_list.email, phone
FROM mailing_list , phone_numbers // NB: Comma for at joine!
WHERE mailing_list.email = phone_numbers.email ;
```

✓ name ✓	✓ email ✓	✗ email	✓ phone ✓
✓ Claus	brabrand@itu.dk	brabrand@itu.dk	7218 5076
✗ Barack	obama@hotmail.com	brabrand@itu.dk	7218 5076
✗ John	jdoe@notmail.com	brabrand@itu.dk	7218 5076
✗ Claus	brabrand@itu.dk	obama@hotmail.com	212-555-0000
✓ Barack	obama@hotmail.com	obama@hotmail.com	212-555-0000
✗ John	jdoe@notmail.com	obama@hotmail.com	212-555-0000
✗ Claus	brabrand@itu.dk	jdoe@notmail.com	123-456-7890
✗ Barack	obama@hotmail.com	jdoe@notmail.com	123-456-7890
✓ John	jdoe@notmail.com	jdoe@notmail.com	

"join" operation:  
alle kombinationer  
ok, nu har vi rækkerne :-)  
og nu også kolonnerne :-)

name	email	phone
Claus	brabrand@itu.dk	7218 5076
Barack	obama@hotmail.com	212-555-0000
John	jdoe@notmail.com	123-456-7890

# Registersammenkøring (Join)

```
SELECT ProductName, CategoryName, Price  
FROM Products , Categories  
WHERE Products.CategoryID=Categories.CategoryID;
```

ProductName	Price	CategoryID	CategoryID	CategoryName	Description
Chais	18	1	1	Beverages	Soft drinks, coffees, teas, beers, and ales
Chang	19	1	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
Aniseed Syrup	10	2	3	Confections	Desserts, candies, and sweet breads
Chef Anton's Cajun Seasoning	22	2			

ProductName	CategoryName	Price
Chais	Beverages	18
Chang	Beverages	19
Aniseed Syrup	Condiments	10
Chef Anton's Cajun Seasoning	Condiments	22

# CSI: Copenhagen



# Eksempel

## Suspects:

name	dna_profile
...	...
Jane Doe	ACTGC
Jerry Doe	ACGTC
John Doe	ACTTC
...	...

## Crimescene:

place	item	dna_found
Stuen	Bord	ACCTC
Stuen	Lampe	AGTGC
Gangen	Kasket	ACCTC
Garagen	Kniv	ACTGC
Badeværelset	Sæbe	ACCTC
Bilen	Lommelygte	ACTTC

```
CREATE TABLE Suspects ( name VARCHAR(20) NOT NULL, dna_profile VARCHAR(10) NOT NULL );
INSERT INTO Suspects (name, dna_profile) VALUES ('Jane Doe', 'ACTGC');
INSERT INTO Suspects (name, dna_profile) VALUES ('Jerry Doe', 'ACGTC');
INSERT INTO Suspects (name, dna_profile) VALUES ('John Doe', 'ACTTC');
SELECT * FROM Suspects;
```

```
CREATE TABLE Crimescene ( place VARCHAR(100) NOT NULL,
                           item VARCHAR(20) NOT NULL,
                           dna_found VARCHAR(20) NOT NULL );
INSERT INTO Crimescene (place,item,dna_found) VALUES ('Stuen','Bord','ACCTC');
INSERT INTO Crimescene (place,item,dna_found) VALUES ('Stuen','Lampe', 'AGTGC');
INSERT INTO Crimescene (place,item,dna_found) VALUES ('Gangen','Kasket', 'ACCTC');
INSERT INTO Crimescene (place,item,dna_found) VALUES ('Garagen','Kniv', 'ACTGC');
INSERT INTO Crimescene (place,item,dna_found) VALUES ('Badeværelset','Sæbe', 'ACCTC');
INSERT INTO Crimescene (place,item,dna_found) VALUES ('Bilen','Lommelygte', 'ACTTC');
SELECT * FROM Crimescene ;
```

# Join



## Suspects:

name	dna_profile
...	...
Jane Doe	ACTGC
Jerry Doe	ACGTC
John Doe	ACTTC
...	...

... + 200.000 flere personer

## Crimescene:

place	item	dna_found
Stuen	Bord	ACCTC
Stuen	Lampe	AGTGC
Gangen	Kasket	ACCTC
Garagen	Kniv	ACTGC
Badeværelset	Sæbe	ACCTC
Bilen	Lommelygte	ACTTC

**Join** til at finde de skyldige blandt de mistænkte?



name	place	item
Jane Doe	Garagen	Kniv
John Doe	Bilen	Lommelygte



# OPGAVE

# Join



## Suspects:

name	dna_profile
...	...
Jane Doe	ACTGC
Jerry Doe	ACGTC
John Doe	ACTTC
...	...

... + 200.000 flere personer

## Crimescene:

place	item	dna_found
Stuen	Bord	ACCTC
Stuen	Lampe	AGTGC
Gangen	Kasket	ACCTC
Garagen	Kniv	ACTGC
Badeværelset	Sæbe	ACCTC
Bilen	Lommelygte	ACTTC

**Join** til at finde de skyldige blandt de mistænkte?

```
SELECT name, place, item  
FROM Suspects , Crimescene  
WHERE dna_profile = dna_found ;
```



name	place	item
Jane Doe	Garagen	Kniv
John Doe	Bilen	Lommelygte



# OPGAVE



# JOIN



**Sammenkør** tabellerne **Products** og **Suppliers**,  
således at de passer sammen på **SupplierID**:

Vis evt blot **ProductName** samt **Phone**:

<b>ProductName</b>	<b>Phone</b>
Alice Mutton	(03) 444-2343
Aniseed Syrup	(171) 555-2222
Boston Crab Meat	(617) 555-3267
Camembert Pierrot	38.76.98.06
Carnarvon Tigers	(03) 444-2343
Chais	(171) 555-2222

Sortér evt efter **ProductName**



# OPGAVE



# SQL Injection Attack

```
<html>
<script>
```

```
function q() {
    var name = document.getElementById('name').value;
    var query = "SELECT * FROM users WHERE name = '" + name + "';";
    alert(query);
    /* contact database with query... */
}
```

<https://xkcd.com/327/>

```
</script>
```

Hvad er dit navn:

Claus

```
<body>
Hvad er dit navn:<br/>
<input id="name"/> ?
<p/>
<button onclick="q();">
    Click her!
</button>
</body>
</html>
```

SELECT \* FROM users WHERE name = 'Claus';

Hvad er dit navn: ?

Claus'; DROP TABLE users; --

SELECT \* FROM users WHERE name = 'Claus'; DROP TABLE users; --';

# **Objekter & Objekt-Orientering**

# JavaScript Objekter

Et *JavaScript Objekt* indkapsler data:

```
var person = {  
    name : "Claus",  
    age  : 51  
};
```

Et *Objekt* har *felte*,  
der svarer til *variabler*

Felterne tilgås med  
*punktum-notation*

Sådan kan man *manipulere objekter*:

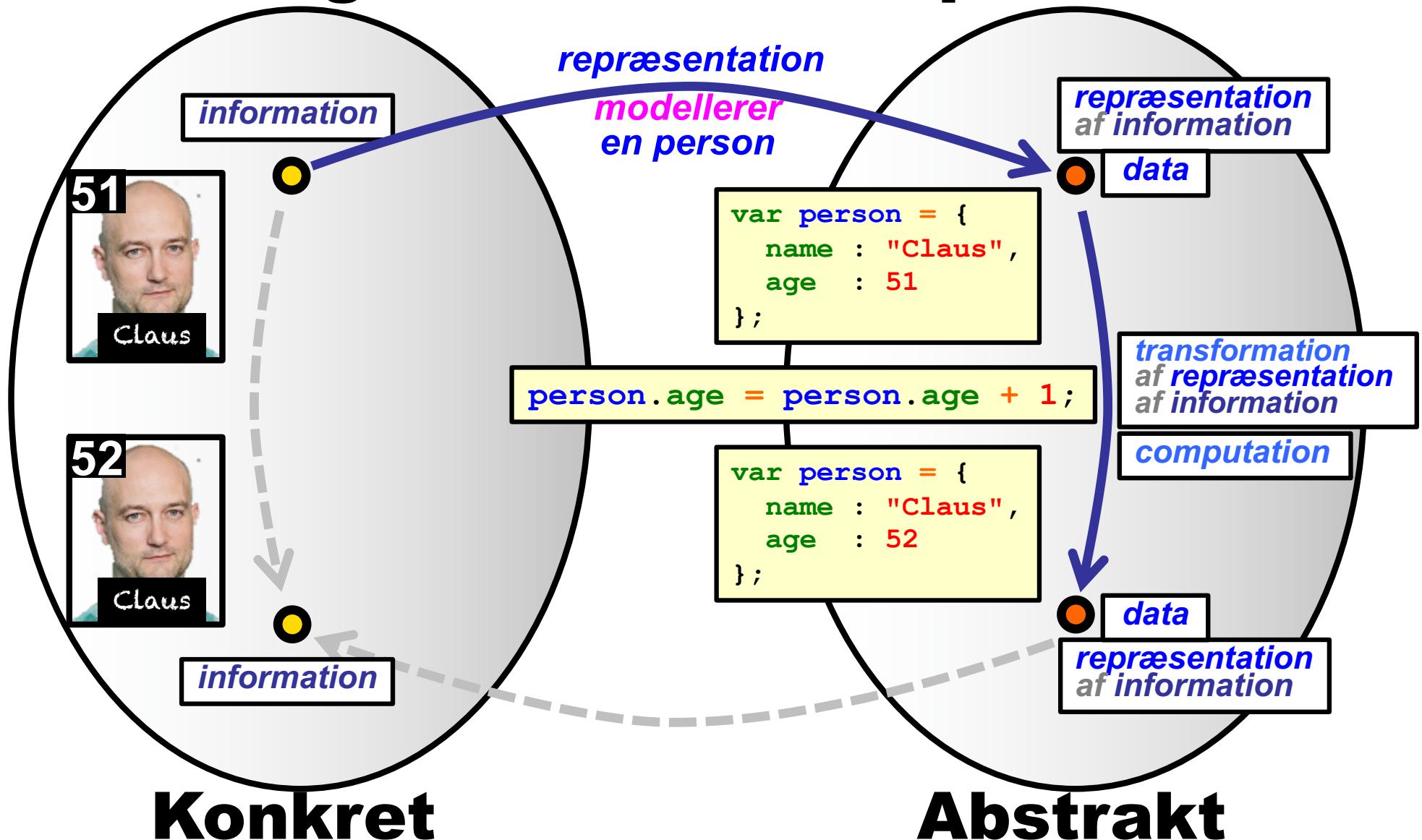
```
document.write('<li>');  
document.write( person.name ); //LÆSE fra felt  
document.write(' (' + person.age + ')</li>'); //LÆSE fra felt  
  
person.age = person.age + 1; // (læse og) SKRIVE til felt  
  
document.write('<li>');  
document.write( person.name );  
document.write(' (' + person.age + ')</li>');
```

- Claus (51)
- Claus (52)

<https://www.w3schools.com/code/tryit.asp?filename=G49TNICWTBP9>

# Repræsentation af Objekter

## Virkelighed Computer-model



# Objekter



```
var person = {  
    name : "Claus",  
    age  : 51  
};
```

- 1) Lav et objekt, der repræsenterer en **bank-transaktion** (dvs fra en person til en anden):

Tænk over hvordan du **repræsenterer informationer**

- 2) Prøv at læse et par af felterne
- 3) Prøv at ændre et par af felterne

<https://www.w3schools.com/code/tryit.asp?filename=G49TNICWTBP9>



## OPGAVE



# Objekter med metoder

## JavaScript Objekt:

```
var person = {  
    name : "Claus",  
    age  : 51  
}
```

```
document.write(person.name);  
document.write(' (' + person.age + ')');
```

Claus (51)

## ...med egne *metoder*:

```
var person = {  
    name      : "Claus",  
    age       : 51,  
    birthday : function() { this.age++; },  
    display   : function() { document.write(this.name + ' (');  
                            document.write(this.age + ')<br/>');  
    }  
};
```

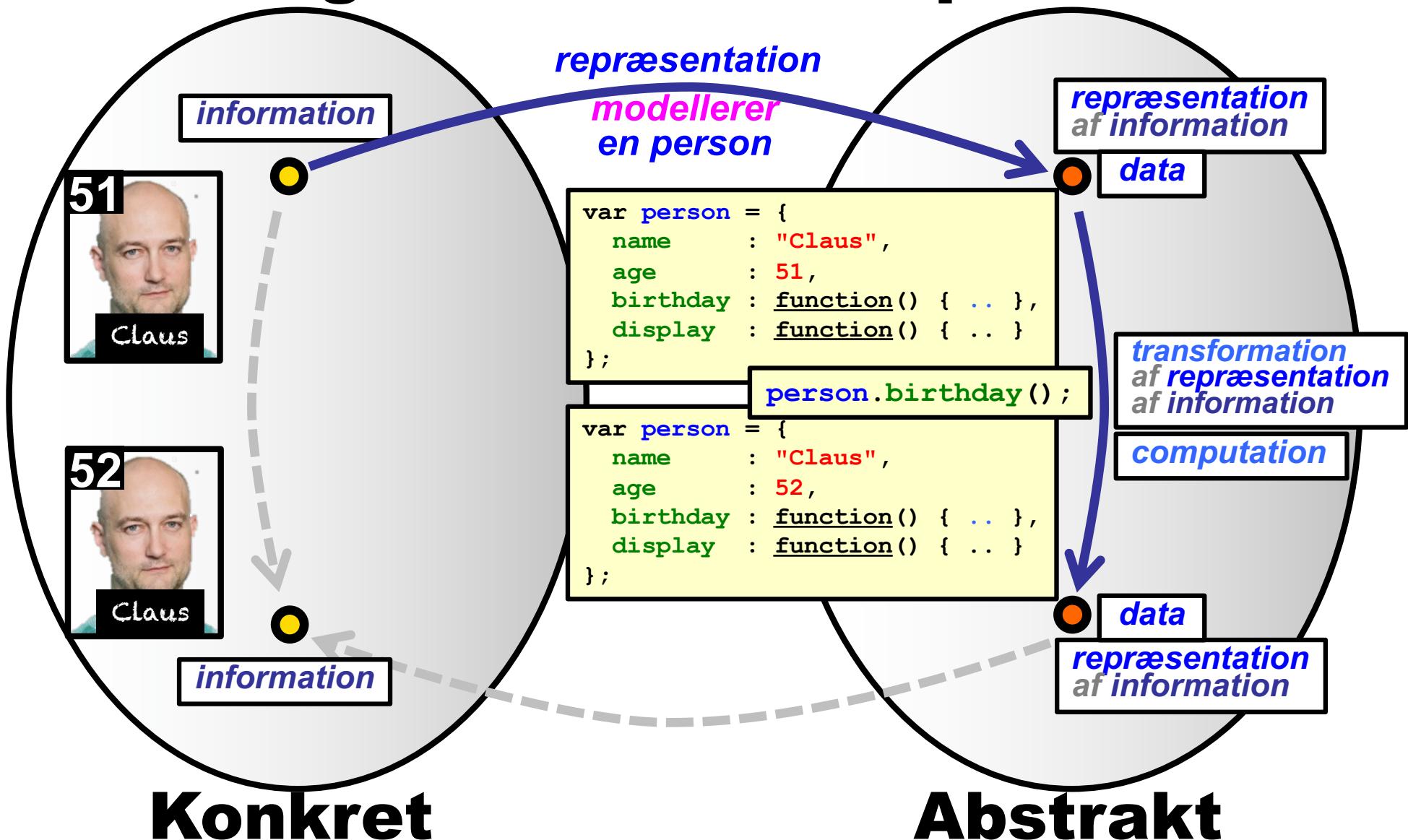
**NB:** Det er sådan  
`write()` er defineret  
på `document` objektet!

```
person.display();  
person.birthday();  
person.display();
```

Claus (51)  
Claus (52)

# Objekt-Orientering

## Virkelighed Computer-model



# **JSON**

## JSON (JavaScript Object Notation):

Simpelt format til udveksling af data på internettet

### *JavaScript Objekt:*

```
var person = {  
    name: "Claus",  
    age: 51  
};
```

### *JSON Repræsentation:*

```
{ "name": "Claus", "age": 51 }
```

NB: "Gåseøjne" rundt om felt-navne (alt undtagen tal)

# JavaScript JSON

## JavaScript → JSON:

```
var person = {  
    name: "Claus",  
    age: 51  
};
```

```
var j = JSON.stringify(person);  
document.write(j);
```

```
{"name": "Claus", "age": 51}
```

<https://www.w3schools.com/code/tryit.asp?filename=G49U5FJ0OEEV>

## JSON → JavaScript:

```
var j = '{ "name": "Claus", "age": 52 }';  
var person = JSON.parse(j);
```

```
document.write("<b>" + person.name + "</b> is <i>");  
document.write(person.age);  
document.write("</i> years old");
```

```
Claus is 52 years old
```

<https://www.w3schools.com/code/tryit.asp?filename=G49U66JVA0IQ>



# JSON



**1)** Lav jeres objekter fra før om til **JSON** og udskriv dem som **JSON** værdier

[ HINT: Brug **JSON.stringify( . . . )** ]

**2)** Ændr i et af felterne i **JSON** værdien

(Pas på I ikke kommer til at slette et vigtigt symbol)

[ HINT: Kopier og ændr værdien ]

**3)** Konvertér herefter **JSON** værdien til et **JavaScript objekt** & udskriv det ændrede felt

[ HINT: Brug **JSON.parse( . . . )** ]



## OPGAVE



# **Eksempler på JSON Data**

## **Turistinformation om KBH:**

<https://www.europeandataportal.eu/data/dataset/guidedanmark-oplevelser-overnatning-aktiviteter-i-hele-danmark>

## **Reklamebærende Byudstyr (KK):**

<https://data.kk.dk/dataset/reklamebaerende-udstyr>

## **Danmarks Statistik (Statistikbank):**

<https://www.dst.dk/da/Statistik/statistikbanken/api>

## **Vejret i Verden:**

<https://www.openweathermap.org>

## **Dataset fra EU (European Data Portal):**

<https://www.europeandataportal.eu/data/dataset>

## **Google Custom Search:**

<https://developers.google.com/custom-search/json-api/v1/overview>

# Nestede Objekter

Ligesom alt andet, kan objekter også *nestes*:

```
var objekt = {  
    name : "Copenhagen",  
    pop : 1308893,  
    country : {  
        name : "Denmark",  
        pop : 5731000,  
        continent : {  
            name : "Europe",  
            pop : 7414000000,  
            planet : {  
                name : "Earth",  
                pop : 7600000000000  
            }  
        }  
    }  
}  
  
var j = JSON.stringify(objekt);  
document.write(j);  
document.write( objekt.country.continent.name );
```

Europe

<https://www.w3schools.com/code/tryit.asp?filename=FSD3AO4K70A5>

# JSON vs XML

JSON er *mere kompakt* end XML:

```
{ "employees" : [  
    { "firstName" : "John" , "lastName" : "Doe" } ,  
    { "firstName" : "Anna" , "lastName" : "Smith" } ,  
    { "firstName" : "Peter" , "lastName" : "Jones" } ] }
```

repræsentation  
af information



```
<employees>  
  <employee>  
    <firstName>John</firstName> <lastName>Doe</lastName>  
  </employee>  
  <employee>  
    <firstName>Anna</firstName> <lastName>Smith</lastName>  
  </employee>  
  <employee>  
    <firstName>Peter</firstName> <lastName>Jones</lastName>  
  </employee>  
</employees>
```

repræsentation  
af information

...og kan afkodes i én linie:

```
obj = JSON.parse(data);
```

# JSON Pretty Printer

Her er en **JSON Pretty Printer**:

En **Pretty Printer** kan formatttere et program pænt (med indlejring)

<http://jsonprettyprint.com>

**Rå JSON:**

```
{"name": "Copenhagen", "pop": 1308893, "country": { "name": "Denmark", "pop": 5731000, "continent": { "name": "Europe", "pop": 7414000000, "planet": { "name": "Earth", "pop": 7600000000000 } } }}
```

**Pæn JSON:**

```
{  
    "name": "Copenhagen",  
    "pop": 1308893,  
    "country": {  
        "name": "Denmark",  
        "pop": 5731000,  
        "continent": {  
            "name": "Europe",  
            "pop": 7414000000,  
            "planet": {  
                "name": "Earth",  
                "pop": 7600000000000  
            }  
        }  
    }  
}
```

# **Protokoller**

## Google: 'protocol define':

"Regler for formaliteter og ceremonier (fx ifm diplomatisk samkvem)"

"Regler for udveksling af data mellem (forskellige enheder i) edb-systemer"

Eksempel: Bestilling af vin på en restaurant  
(følger klient-server modellen):

Klienten kalder på serveren:                   'Jeg vil vin!'

Serveren foretager en forespørgsel:       'Hvilken vin?'

Klienten kommer med en respons:           'Margaux '80!'

Serveren serverer en smagsprøve:       'Er vinen ok?'

Klienten prøvesmager og responderer: 'Vin'en er ok!'

Serveren serverer alle og afslutter:      'Værsgo: vin!'

...



# Mange måder at sende data på



radio  
110010101001110101

## Herudover:

- Lyd (sonar, vibrationer, hoste)
- Røg / ild (som i "Return of the King")
- Duftsignaler (feromoner)
- Molekyler (DNA / RNA / proteiner)
- Tyngdebølger?
- Cykelbud med portable hardiske
- ...



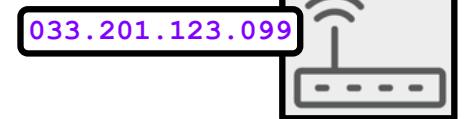
radio  
1010101001110101



elektricitet  
φ→



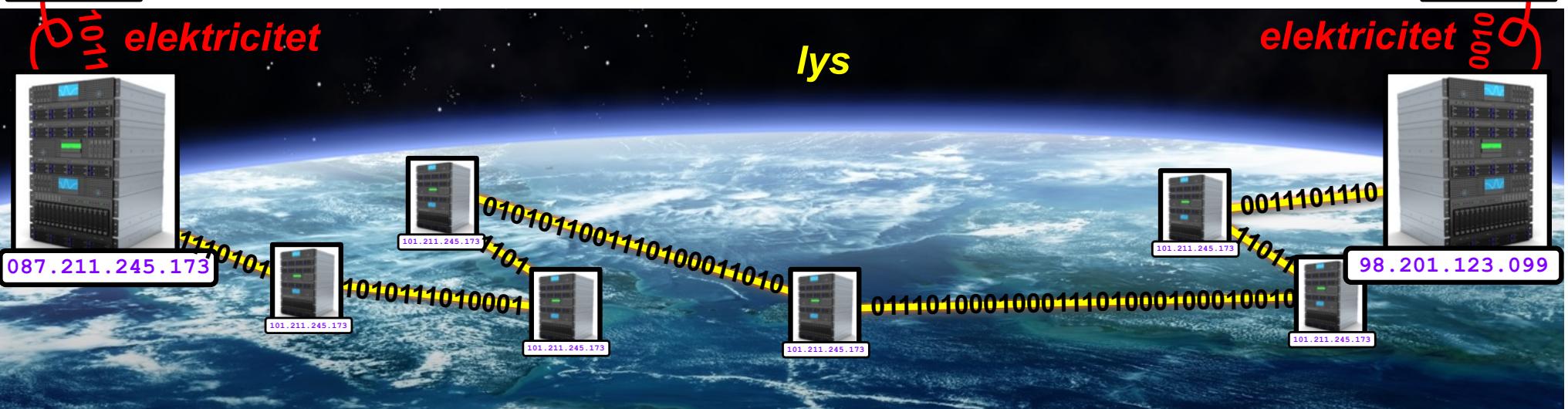
087.211.245.173



elektricitet  
→φ



98.201.123.099



# Lag på Lag af Protokoller



## HTTPS (HyperText Transfer Protocol Secure):

Website autentificeres [vha sikkerhedscertifikater] tu.dk  
1) data krypteres, 2) sendes [vha nedenstående], 3) dekrypteres

## HTTP (HyperText Transfer Protocol):

Introducerer *request* og *respons* der sendes [vha nedenstående]  
En klient kan submitte data til server (fx "google.com?q=itu")

## TCP (Transport Control Protocol):

Store pakker splittes i små pakker der sendes [vha nedenstående] med et sekvens#; modtageren samler alt igen i rigtig rækkefølge ved at bruge sekvens#et (evt tabte pakker re-transmitteres).

## IP (Internet Protocol):

IP-adresser ([101.211.245.173](http://101.211.245.173)) identificerer enheder på nettet  
Små pakker (datagrammer) sendes [vha nedenstående] fra en enhed til en anden enhed (bemærk: pakker kan gå tabt!)

## Binær Data Transmission:

Sende bits (data) kun direkte fra sender til modtager

## "How the Internet Works":

### How the Internet Works

Learn

-  Wires, cables, and WiFi
-  IP addresses and DNS
-  Packet, routers, and reliability
-  HTTP and HTML
-  Encryption and public keys
-  Cybersecurity and crime

<https://www.khanacademy.org/partner-content/code-org/internet-works>

# **Web Services**

(aka, Web Apps / Web Applications)

hjemmesider + programmering + databaser

# Kommunikation (HTTP → JSON)

```
<html><body><script>  
function request(s) {  
    var x = new XMLHttpRequest();  
    x.open('GET', s, false);  
    x.send();  
    return x.responseText;  
}  
  
var json = request("https://ipinfo.io/json");  
  
document.write('<p>Respons fra serveren (i JSON format): </p>');  
document.write(json);  
</script></body></html>
```



ipinfo.io/json

Respons fra serveren (i JSON format):

```
{ "ip": "109.57.10.232", "hostname": "109.57.10.232.mobile.3.dk", "city":  
"Copenhagen", "region": "Capital Region", "country": "DK", "loc": "55.6759,12.5655",  
"org": "AS44034 Hi3G Access AB", "postal": "0917", "timezone": "Europe/Copenhagen",  
"readme": "https://ipinfo.io/missingauth" }
```

<https://www.w3schools.com/code/tryit.asp?filename=G49UBF5PUNWW>



# HTTP → JSON



```
<html><body><script>  
function request(s) {  
    var x = new XMLHttpRequest();  
    x.open('GET', s, false);  
    x.send();  
    return x.responseText;  
}  
  
var json = request("https://ipinfo.io/json");  
var info = JSON.parse(json);  
// OPGAVE: Udskriv ip-adresse og by' !  
  
</script></body></html>
```



ipinfo.io/json

Din IP-adresse er: **109.57.10.232**  
Du kommer fra **Copenhagen**

Respons fra serveren (i JSON format):

```
{ "ip": "109.57.10.232", "hostname": "109.57.10.232.mobile.3.dk", "city":  
"Copenhagen", "region": "Capital Region", "country": "DK", "loc": "55.6759,12.5655",  
"org": "AS44034 Hi3G Access AB", "postal": "0917", "timezone": "Europe/Copenhagen",  
"readme": "https://ipinfo.io/missingauth" }
```

<https://www.w3schools.com/code/tryit.asp?filename=G49UBF5PUNWW>



## OPGAVE



# Vejrudsigt

## Copenhagen Weather Forecast

Hovedstaden, DK

[VIEW ON MAP](#)

[FEEDBACK](#)

[History](#) | [Climate Averages](#) | [14 Day Weather](#)

⌚ Sunrise: 07:01 AM ⛩ Sunset: 07:02 PM 🌙 Moonrise: 07:39 PM 🌙 Moonset: 06:55 AM ⚜ Phase: Full Moon  with 97 % illumination.

### Copenhagen Current weather report

COPENHAGEN 

Local Time: Tue 25<sup>th</sup> Sep 11:57 am



13 °C

Partly cloudy

Wind: 19 km/h from NW 



Rain:

52 %



Humidity:

54%



Pressure:

1034 mb



Cloud:

25%



Visibility:

10 km

<https://www.worldweatheronline.com/copenhagen-weather/hovedstaden/dk.aspx>

# Kommunikation (http → html)

```
<html><body><script>
function request(s) {
    var x = new XMLHttpRequest();
    x.open('GET', s, false);
    x.send();
    return x.responseText;
}

var html = request('https://www.worldweatheronline.com/copenhagen-weather/hovedstaden/dk.aspx');

alert(html);
</script></body></html>
```



worldweatheronline.com

```
<!DOCTYPE html>
<html itemScope itemType="http://schema.org/Organization">
<head id="ctl00_Head1" dir="auto"><meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" />
<title>
    Copenhagen Weather Forecast, Hovedstaden, Denmark | 10 Day Weather Forecast
</title><meta http-equiv="Content-Type" content="text/html; charset=utf-8" /><meta id="ctl00_MetaDataTag" name="description" content="Copenhagen, Denmark Current weather report, Copenhagen, Denmark, Hovedstaden, Denmark Hour by hour 10 Day Weather Forecast, Historical Weather, Weather Animated Map, holiday weather forecast and more." /><meta
```

id="ctl00\_MetaDataTag" name="description" content="Copenhagen, Denmark Current weather report, Copenhagen, Denmark, Hovedstaden, Denmark Hour by hour 10 Day Weather Forecast, Historical Weather, Weather Animated Map, holiday weather forecast and more." /><meta

Close

<https://www.w3schools.com/code/tryit.asp?filename=G49UA53HKDG8>

# Web Scraping

```
<html><body><script>
function request(s) {
    var x = new XMLHttpRequest();
    x.open('GET', s, false);
    x.send();
    return x.responseText;
}

var html = request('https://www.worldweatheronline.com/copenhagen-weather/hovedstaden/dk.aspx');

var s = html.match(<div class="report_text temperature" style="color:#[A-Z0-9]+;\">([0-9]+)</div>')[1];

document.write('Vejret i København er: <b>');
document.write(s);
document.write('&deg;C</b>');
</script></body></html>
```



worldweatheronline.com

**regexp**

Vejret i København er: 6 ° C

<https://www.w3schools.com/code/tryit.asp?filename=G400GWJKEU0H>

# Vejret i København

## Data i JSON format fra Open Weather Map\*:

<https://www.openweathermap.org>

\* registrér (gratis!) og få  
jeres egen APPid nøgle

```
<html><body><script>
function request(s) {
    var x = new XMLHttpRequest();
    x.open('GET', s, false);
    x.send();
    return x.responseText;
}
var city = 'Copenhagen';
var key = 'ddc97e83bd1567b6b1127ebe6b6baa09'; // brug jeres egen nøgle

var json = request('https://api.openweathermap.org/data/2.5/weather?q=' +
                    city + '&units=metric&APPID=' + key);
var result = JSON.parse(json);

document.write('Vejret i ' + city + ' er: <b>');
document.write(result.main.temp);
document.write('&deg;C</b>');
</script></body></html>
```



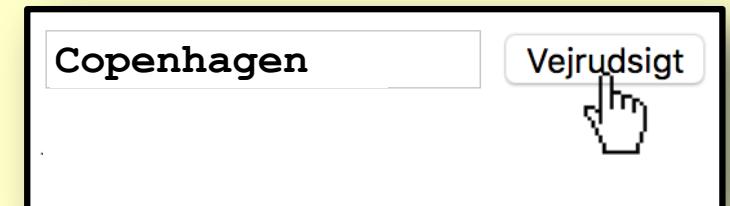
[api.openweathermap.org](https://api.openweathermap.org)

Vejret i Copenhagen er: 4.32°C

<https://www.w3schools.com/code/tryit.asp?filename=G49UCOOY7LKR>

# Vejret i ...

```
<html>
  <head><meta charset="UTF-8"/></head>
  <body>
    <input id="city" value="Copenhagen"/>
    <button onclick="
function request(s) {
  var x = new XMLHttpRequest();
  x.open('GET', s, false);
  x.send();
  return x.responseText;
}
var city = document.getElementById('city').value;
var key = 'ddc97e83bd1567b6b1127ebe6b6baa09'; // brug jeres egen nøgle
var json = request('https://api.openweathermap.org/data/2.5/weather?q=' +
  city + '&units=metric&APPID=' + key);
var result = JSON.parse(json);
document.getElementById('vejr').innerHTML = 'Vejret i ' + city + ' er: <b>' +
  result.main.temp + '&deg;C</b>';
"
    >
    Vejrudsigt
  </button>
  <p id="vejr"></p>
</body>
</html>
```



<https://www.w3schools.com/code/tryit.asp?filename=G49UEIOX0R7Y>

# Opdaterede Valutakurser

```
<html>
  <head><meta charset="UTF-8"/></head>
  <body>
    <button onclick="
function request(s) {
  var x = new XMLHttpRequest();
  x.open('GET', s, false);
  x.send();
  return x.responseText;
}

var json = request('https://www.floatrates.com/daily/dkk.json');
var result = JSON.parse(json);
var dkk2eur = result.eur.rate;
var eur2dkk = result.eur.inverseRate; // or: eur2dkk = 1 / dkk2eur
document.getElementById('kurs').innerHTML =
'1 dkk = <b>' + dkk2eur + '</b> eur<br/>' +
'1 eur = <b>' + eur2dkk + '</b> dkk';
      Kurs
    </button>
    <p>
      <code id="kurs"></code>
    </body>
  </html>
```

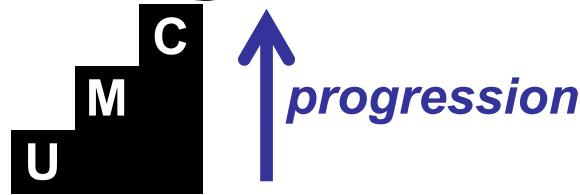
Kurs

1 dkk = 0.1341296644571 eur  
1 eur = 7.4554723151482 dkk

<https://www.w3schools.com/code/tryit.asp?filename=G49UEZYA0UB5>

**Hvordan-kommer-jeg-selv-videre-med-programmering?**

# Use-Modify-Create



## 1) Brug:

Først: Afprøv det der bliver introduceret; **kør** (brug) programmet og se hvad der sker.

## 1) Use

## 2) Modificér:

Dernæst: Prøv at **modificere** programmet de interessante og relevante steder og se hvad der sker når man ændre lidt på den forventede opførsel. I dette skridt kan man afprøve en hel masse ændringer en ad gangen og på den måde få overblik over samt erfaring med de konstruktioner.

## 2) Modify

## 3) Skab:

Sidst: Prøv at skrive et program (evt næsten) fra bunden der gør et-eller-andet man har tænkt sig på forhånd (på samme måde som vi gjorde med Tæl-Antal-a'er). Dette er det sværreste skridt. Her er udfordringen ikke at lave det for svært/ambitiøst. #BlinkendeCursorØversteVenstreHjørne

## 3) Create

# w3school Online Try-It Service



# JavaScript Tutorial

w3schools.com

THE WORLD'S LARGEST WEB DEVELOPER SITE



HTML

CSS

JAVASCRIPT

SQL

PHP

MORE ▾

EXAMPLES ▾



JS Tutorial

JS HOME

JS Introduction

JS Where To

JS Output

JS Statements

JS Syntax

JS Comments

JS Variables

JS Operators

JS Arithmetic

JS Assignment

JS Data Types

JS Functions

JS Objects

JS Scope

JS Events

JS Strings

JS String Methods

JS Numbers

JS Number Methods

JS Arrays

JS Array Methods

JS Array Sort

.NET Excel

Excel R/W, Lightning Fast, Easy Free Support, Upgrades & Try! e-iceblue.com



Cut Development  
Time in Half With  
Telerik UI for  
ASP.NET MVC

## JavaScript Introduction

◀ Previous

Next ▶

This page contains some examples of what JavaScript can do.



Try Now

## JavaScript Can Change HTML Content

Progress®

One of many JavaScript HTML methods is **getElementById()**.

This example uses the method to "find" an HTML element (with id="demo") and changes the element content (**innerHTML**) to "Hello JavaScript":

### Example

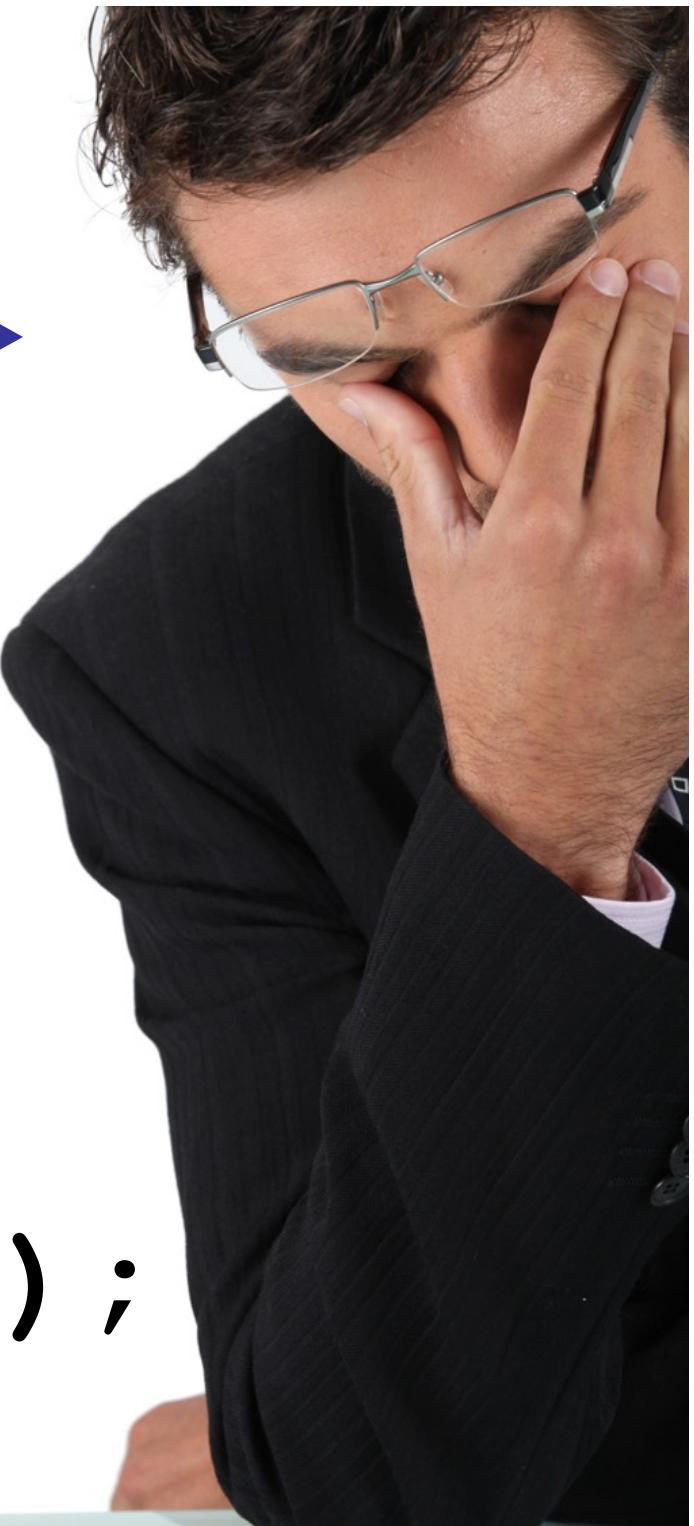
```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

</kursus>

?q=spørgsmål

// kommentarer

kursus.klager();



# Kursusindhold

## HTML

Attributter  
CSS

## HTML FORMS

Input felter

## JavaScript:

Konstanter  
Variabler  
Tildeling  
Operatorer  
Brugerinput  
Output  
if/while/for  
switch  
funktion  
objekt

## Objekt-orientering

## JSON

JSON Format

## Databaser

SQL

## Protokoller

## Web Services:

## IT-Tankegang!

```
<html><body><table><tr><td>...</td></tr></table>...
<font color="red">...</font>
<h1 style="background-color: red;"></h1>
<form action="url"> <input type="radio" name="x"/>
text, password, radio, checkbox, submit, reset, select, hidden, ...
<script>...</script>    <button onclick="...">click</button>
42, "hej med dig!", true, ...
var x = 42;  var y = "hej";  var z = true;
x = 87;  y = "hello";  z = false;
x = y + z;
prompt("indtast et tal");  document.getElementById('x').value
document.write(..); alert(...); document.getElementById('x').innerHTML = ...;
...
...
function bmi(h,v) { ... return x; }
var person = { name: "Claus", age: 46 };
```

## JavaScript ↔ JSON

```
{ "name": "Claus", "age": 46 }
```

## SQL

SELECT-FROM-WHERE, JOIN, INSERT, UPDATE, DELETE, ...

lag: binær data, IP, TCP, HTTP, HTTPS

http → [service] → JSON

...

**Tak**

Spørgsmål? Kommentarer? Klager?

**TAK  
FOR  
ALT**

# **BONUS SLIDES**

# **Output**

3 måder at lave output på i HTML

# Output (3x)

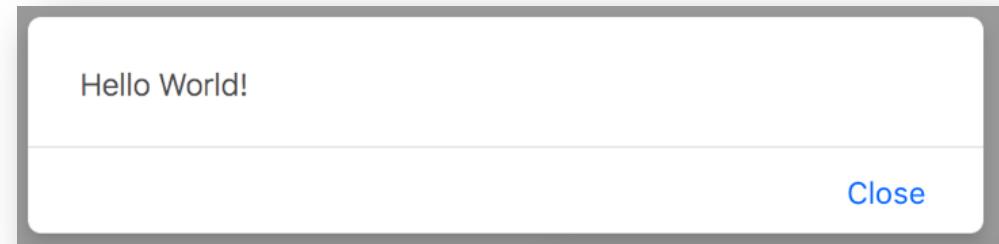
## Skrivning i dokument:

```
<script>
  document.write("Hello World !");
</script>
```



## Alert:

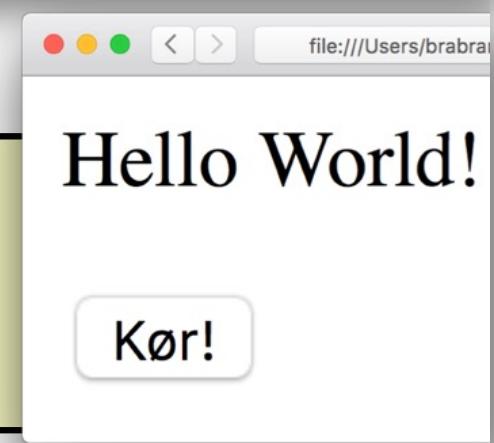
```
<script>
  alert("Hello World !");
</script>
```



## Modificering af dokument:

```
<p id="demo"></p>
<button onclick="document.getElementById('demo')
  .innerHTML = 'Hello World!';">
Kø!</button>
```

Nyt indhold tilføjes til dokumentet lige her når knappen klikkes!





# Output (3x)



Afprøv de *tre forskellige output* muligheder:

## 1) Skrivning i dokument:

<http://www.itu.dk/people;brabrand/ITU-PC/output1.html>

<https://www.w3schools.com/code/tryit.asp?filename=FRVV2N5BH900>

## 2) Alert:

<http://www.itu.dk/people;brabrand/ITU-PC/output2.html>

<https://www.w3schools.com/code/tryit.asp?filename=FRVV3PNVACAO>

## 3) Modificering af dokument:

<http://www.itu.dk/people;brabrand/ITU-PC/output3.html>

<https://www.w3schools.com/code/tryit.asp?filename=FRVV4HT8R8OU>

Prøv: at lave om på hvert af de tre programmer så der i stedet udskrives et-eller-andet andet :-)



# OPGAVE



# Output (3x)

## 1) Skrivning i dokument:

`document.write(...)` benyttes kun **før** dokumentet er færdigskrevet (dvs aldrig ifm fx `onclick`)

## 2) Alert:

`alert(...)` kan altid bruges – både **før** og **efter** dokumentet er færdigskrevet.

- (Bruges mest ifm fejl og debugging.)

## 3) Modificering af dokument:

`.innerHTML` benyttes kun **efter** dokumentet er færdigskrevet; dvs ifm fx `onclick`

# **Web Services**

# SQL + JavaScript

```
<html><body><script>

function request(s) {
    var x = new XMLHttpRequest();
    x.open('GET', s, false);
    x.send();
    return x.responseText;
}

var json = request('https://www.w3schools.com/js/customers_mysql.php');

var customers = JSON.parse(json);

document.write('<table border="1">');
document.write('<tr><th>Name</th><th>City</th><th>Country</th></tr>');
for (let customer of customers) {
    document.write('<tr>');
    document.write('<td>' + customer.Name + '</td>');
    document.write('<td>' + customer.City + '</td>');
    document.write('<td>' + customer.Country + '</td>');
    document.write('</tr>');
}
document.write('</table>');
</script></body></html>
```

Name	City	Country
Alfredo Futterkiste	Berlin	Germany
Berglunds snabbköp	Luleå	Sweden
Centro comercial Moctezuma	México D.F.	Mexico
Erns Handel	Graz	Austria
FISSA Fabrica Inter. Salchichas S.A.	Madrid	Spain
Galería del gastrónomo	Barcelona	Spain
Island Trading	Cowes	UK
Königlich Essen	Brandenburg	Germany
Laughing Bacchus Wine Cellars	Vancouver	Canada
Magazzini Alimentari Riuniti	Bergamo	Italy
North/South	London	UK
París spécialités	Paris	France
Rattlesnake Canyon Grocery	Albuquerque	USA
Simoni bistro	København	Denmark
The Big Cheese	Portland	USA
Vaffeljernet	Århus	Denmark
Wolski Zajazd	Warszawa	Poland

<https://www.w3schools.com/code/tryit.asp?filename=G49UCAUJ1K41>

# **Lister**

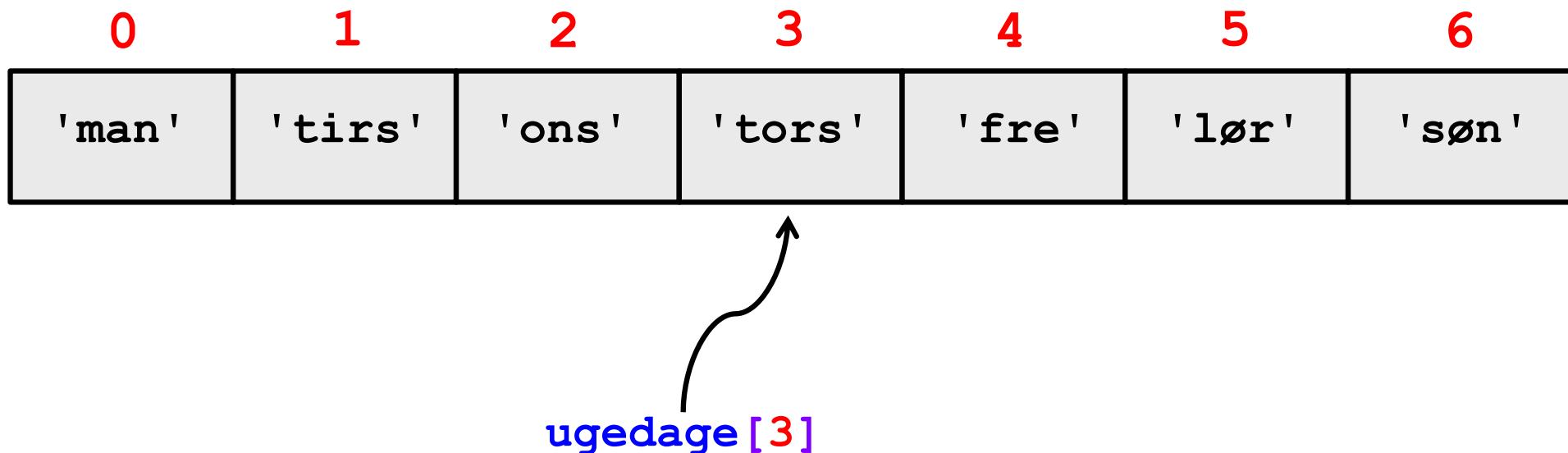
## Arrays

# Repræsentation

Konstant:

```
var ugedage = [ 'man' , 'tirs' , 'ons' , 'tors' , 'fre' , 'lør' , 'søn' ];
```

Repræsentation:



# Lister (Arrays)

## Konstant:

```
var ugedage = [ 'man', 'tirs', 'ons', 'tors', 'fre', 'lør', 'søn' ];
```

## Længde:

```
document.write('Der er <b>' + ugedage.length + '</b> ugedage<br/>');
```

## Tilgang (get value):

```
document.write('Første ugedag er: <b>' + ugedage[0] + '</b><br/>');
```

```
document.write('Sidste ugedag er: <b>' + ugedage[?] + '</b><br/>');
```

## Tildeling (set value):

```
ugedage[3] = 'Thors'; // set  
document.write(ugedage[3] + 'dag'); // get
```

<https://www.w3schools.com/code/tryit.asp?filename=FVCO2XS02XE6>

# Gennemløb (vha 'for')

Konstant:

```
var ugedage = [ 'man' , 'tirs' , 'ons' , 'tors' , 'fre' , 'lør' , 'søn' ] ;
```

Gennemløb (vha 'for'):

```
document.write('<ul>') ;
for ( var index = 0 ; index < ugedage.length ; index++ ) {
    document.write('<li>' + ugedage[index] + 'dag</li>');
}
document.write('</ul>');
```

<https://www.w3schools.com/code/tryit.asp?filename=FVDXQJQ34RFP>

# Nyt Array

## Konstant:

```
var ugedage = [ 'man', 'tirs', 'ons', 'tors', 'fre', 'lør', 'søn' ];
```

## Er det samme som:

```
var ugedage = new Array(7);
// Bemærk: 'ugedage' er nu et array med 7 (syv) 'undefined' værdier!

ugedage[0] = 'man';
ugedage[1] = 'tirs';
ugedage[2] = 'ons';
ugedage[3] = 'tors';
ugedage[4] = 'fre';
ugedage[5] = 'lør';
ugedage[6] = 'søn';
```

# **Input Validering**

# Regulære Udtryk (regexps)

Regexp:

$R \rightarrow c \mid RR \mid R|R \mid R^*$

Bemærk:  
blot **fire** (4)  
konstruktioner!

Enkeltkarakter: c

Matcher: blot "sig selv"

Sekvens:  $R_1R_2$

Matcher: regexp  $R_1$  **efterfulgt af** regexp  $R_2$

Eksempel: "june|july|august" sommermåneder!

Alternativ:  $R_1 | R_2$

Matcher: regexp  $R_1$  **eller** regexp  $R_2$

Eksempel: regexp "a|e|i|o|u" matcher en **vokal!**

Opgave: skriv et regexp der matcher et **chifer!**

# Regulære Udtryk (regexps)

Regexp:

$R \rightarrow c \mid RR \mid R|R \mid R^*$

Bemærk:  
blot **fire** (4)  
konstruktioner!

Parantes:

$(R)$

Matcher: regexp  $R$

Eksempel: "ju(ne|ly)|august" sommermåneder!

Opgave: match et *postnummer*!

Repetition:

$R^*$

Matcher:  $R$  efterfulgt af sig selv **nul-eller-flere gange**

Eksempel: "(0|1|2|3|4|5|6|7|8|9)\*"

Opgave: forbedr ovenstående, så det ikke inkluderer den tomme streng og så man ikke kan skrive angive foranstillede nuller (som i for eksempel "01" el "007")

# Forkortelser

**Chifer:**  $\backslash\text{d}$   $\equiv$  0|1|2|3|4|5|6|7|8|9

**Karaktermængde:** [aeiou]  $\equiv$  a|e|i|o|u

**Karakterinterval:** [a-z]  $\equiv$  a|b|c|d|...|z

**Valgfri:** R?  $\equiv$  (R|)

**En-eller-flere:** R+  $\equiv$  RR\*

Opgave: hvad matcher "\d+" ?

**Repetitioner:**

R{3}  $\equiv$  RRR

R{2,3}  $\equiv$  RRR?

R{2,}  $\equiv$  RRR\*

# Regulære Udtryk (regexps)

			regexp kerne!
c	enklet karakter	x	
RR	konkatenering	abekat	
R R	alternativer	dig mig	
R*	nul-eller-flere	10*	
(R)	parentes (præcedens override)	(d m) ig	
\d	chifer (digit)	\d = (0 1 2 3 4 5 6 7 8 9)	
R+	en-eller-flere	R+ = RR*	
R?	valgfri	R? = ( R)	
R{n}	n gentagelser	R{3} = RRR	
R{n,m}	n-til-m gentagelser	R{2,3} = RRR?	
R{n,}	n-eller-flere	R{2,} = RRR*	
[a-z]	fra 'a' til 'z'	[a-e] = (a b c d e)	
[...]	vilkårlig karakter i mængden (a e i o u)	[aeiou] =	
[^...]	vilk. karakter <i>ikke</i> i mængden		

# Eksempler (regexp)

Postnummer:

[0-9] {4}

Kreditkort:

[0-9] {13,16}

Tal (uden foranstillet nul):

0 | [1-9] [0-9] \*

Datoer (simplificeret):

[0-9] {2} / [0-9] {2} / [0-9] {4}

Datoer (lidt bedre):

(0[1-9] | [1-2][0-9] | 3[01]) / (0[1-9] | 1[0-2]) / [0-9] {4}

# Input Validering

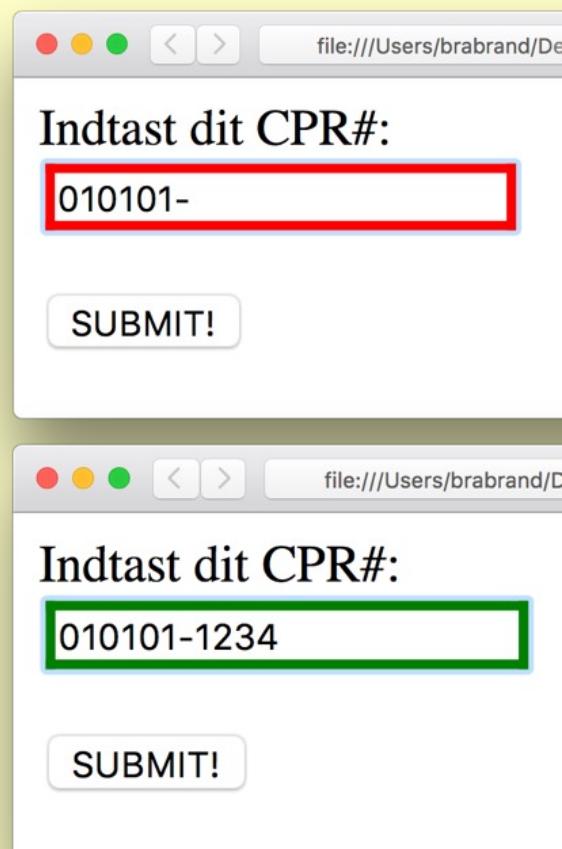
```
<html>
  <head>
    <style>
      input#c { border: 3px solid #eee; }
      input#c:valid { border-color: green; }
      input#c:invalid { border-color: red; }
    </style>
  </head>
  <body>
    <form action="http://www.brics.dk/ixwt/echo">
      Indtast dit CPR#:<br/>
      <input type="text" name="cpr" id="c" required="required" pattern="[0-9]{6}-[0-9]{4}" />
      <p/>
      <input type="submit" value="SUBMIT!" />
    </form>
  </body>
</html>
```

**CSS**

**regexp**

[https://www.w3schools.com/Tags/att\\_input\\_pattern.asp](https://www.w3schools.com/Tags/att_input_pattern.asp)

<https://www.w3schools.com/code/tryit.asp?filename=FV99F1N6F81Q>





# Regulære Udtryk



<https://www.w3schools.com/code/tryit.asp?filename=FV99F1N6F81Q>

## ■ 1) Komma-tal:

- '1.1', '22.22', '333.333' ✓
- '42', '1.2.3', '.5' ✗

## ■ 2) Tidspunkter:

- '00:00', '10:45', '23:59' ✓
- '8:00', '12:60', '99:99' ✗

## ■ 3) IPv4-adresser:

- '0.0.0.0', '65.127.19.221', '255.255.255.255' ✓
- '256.0.0.0', '101.102.103', '1.2.3.4.5' ✗

## ■ 4) Email adresser:

- 'brabrand@itu.dk' ✓
- 'xyz@aaa.bbb.ccc.ddd' ✓
- 'xyz@aaa.bbb.ccc.ddd.' ✗

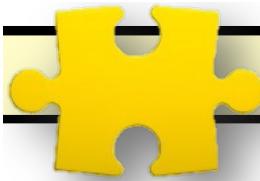


## OPGAVE



# **Spil**

Programmering af Spil  
(Udfordring ★)



# Gæt et Tal (★)



```
<html>
  <head><meta charset="UTF-8"/></head>
  <body onload="

var secret = Math.floor(100 * Math.random()); // tilfældigt tal mellem 0 og 99
var tries = 0;
var guess = -1;
var answer = 'Indtast et tal mellem 0 og 99...';

// WHILE: så længe 'guess' ikke er det samme som 'secret', så bliv ved med at:

  guess = prompt( answer ); // afkræv brugeren et gæt!

  // opdater antal 'tries'

  // sæt 'answer' til hvorvidt 'guess' var 'for højt/lavt' ift 'secret'

// END WHILE (dvs 'guess' må være blevet det samme som 'secret').

// 'alert' til brugeren hvor mange 'tries' blev brugt (& evt hvad 'secret' var)
  ">

  </body>
</html>
```



## OPGAVE



# Gæt et Tal

```
<html>
  <head>
    <meta charset="UTF-8"/>
  </head>
  <body onload="
var secret = Math.floor(100 * Math.random()); // tilfældigt tal mellem 0 og 99
var tries = 0;
var guess = -1;
var answer = 'Indtast et tal mellem 0 og 99...';

while ( guess != secret ) {
  guess = prompt( answer );
  tries = tries + 1;
  if ( guess > secret ) {
    answer = 'for højt!';
  } else if ( guess < secret ) {
    answer = 'for lavt!';
  }
}
alert('Svaret var: ' + secret + '. Du brugte ' + tries + ' gæt.')">

  </body>
</html>
```

<https://www.w3schools.com/code/tryit.asp?filename=G49UPBRCOASA>

# Gæt et Tal

```
<html><head><meta charset="UTF-8"/></head>
<body onload="

var secret = Math.floor(100 * Math.random()); // tilfældigt tal mellem 0 og 99
var tries = 0;
var guess = -1;
var answer = 'Indtast et tal mellem 0 og 99...';

while ( guess != secret ) {
    guess = prompt( answer );
    tries = tries + 1;
    if ( guess > secret ) {
        answer = 'for højt!';
    } else if ( guess < secret ) {
        answer = 'for lavt!';
    }
    document.getElementById('guesses').innerHTML
        += '<li>' + guess + ' var ' + answer + '</li>';
}
alert('Svaret var: ' + secret + '. Du brugte ' + tries + ' gæt.')">
```

```
<ol id="guesses"></ol>
</body>
</html>
```

<https://www.w3schools.com/code/tryit.asp?filename=G49UNSB1QBPR>

# Alternativ Løsning (uden while)

```
<html><head><meta charset="UTF-8"/></head><body>
<script>
    var secret = Math.floor(100 * Math.random());
    var tries = 0;
</script>

Gæt et tal mellem 1 og 100:<br/>
<input id="g"/>

<button onclick="
    var guess = document.getElementById('g').value;
    var answer;
    tries = tries + 1;
    if ( guess > secret ) {
        answer = '<font color=\'red\'>too high</font>';
    } else if ( guess < secret ) {
        answer = '<font color=\'orange\'>too low</font>';
    } else {
        answer = '<font color=\'green\'>right!!!</font>';
    }
    document.getElementById('guesses').innerHTML
        += '<li>' + guess + ' var ' + answer + '</li>';">
    Gæt!
</button>

<ol id="guesses"></ol>
</body></html>
```

<https://www.w3schools.com/code/tryit.asp?filename=FS5008P98J52>



# Spillet NIM (★★)



<http://itu.dk/people/brabrand/nim.html>

Velkommen til spillet NIM

Spiller 1: hvor mange tændstikker vil du tage (1--3)?

- Der er 15 tændstikker på bordet
- Spiller 1 tog 3 fra bordet (12 tilbage)
- Spiller 2 tog 2 fra bordet (10 tilbage)



NIM er et spil for **to personer**. Spillet starter med at der er et antal tændstikker på et bord mellem de to spillere.

Spillerne skiftes nu til at tage mellem en og tre tændstikker (dvs 1, 2 eller 3 tændstikker). Den der tager den **sidste tændstik** fra bordet har **vundet** spillet.

Man **taber** spillet, hvis man tager et **for mange tændstikker** (dvs *flere* end der er tilbage på bordet).

Tryk på 'reload' for at spille igen.



## OPGAVE



```

<html>
  <head>
    <meta charset="UTF-8"/>
  </head>

  <body onload="gentegnBordet(); ">
    <script>
```

HTML

```

      var tilbage = 15;
      var spiller = 1;
      var status = 'Der er ' + tilbage + ' tændstikker på bordet';

      function nim(antal) { /* = HER STÅR DET MESTE AF PROGRAMMERINGEN (se næste slide!) */ }
    </script>
```

JavaScript

<h3>Velkommen til spillet NIM</h3>

Spiller <b id="spiller">1</b>: hvor mange tændstikker vil du tage (1--3)?<br/>

```

<table width="450">
  <tr>
    <td align="center"><button onclick="nim(1); ">En (1)</button></td>
    <td align="center"><button onclick="nim(2); ">To (2)</button></td>
    <td align="center"><button onclick="nim(3); ">Tre (3)</button></td>
  </tr>
</table>
```

```

<ul id="status">
  <li><script>document.write(status); </script></li>
</ul>
```

```

<p id="bordet"></p>
</body>
</html>
```

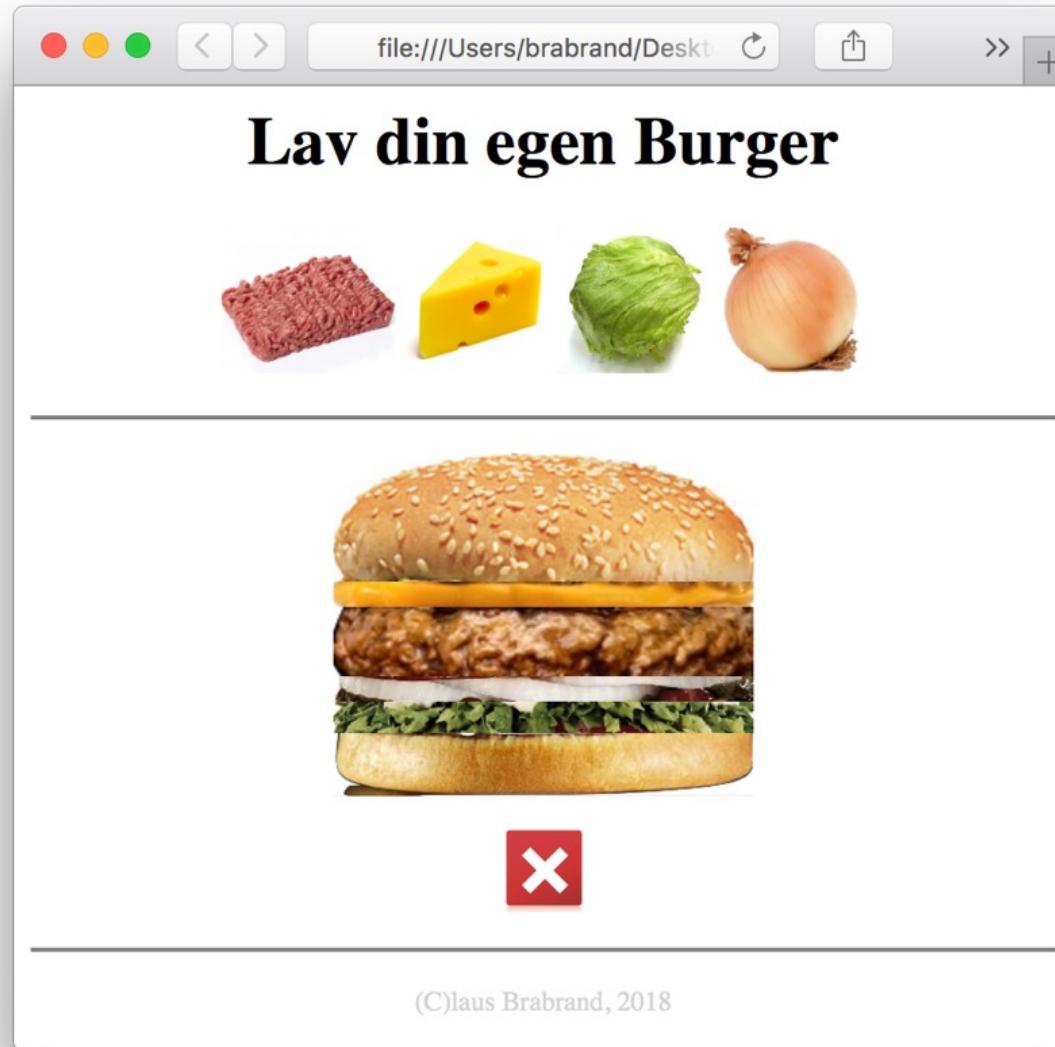
# Nim (fortsat)

JavaScript

```
function gentegnBordet() {
    var s = ' ';
    for ( var i = 0 ; i < tilbage ; i++ ) {
        s += ' ';
    }
    document.getElementById('bordet').innerHTML = s;
}

function nim(antal) {
    if ( antal > tilbage ) {
        status = 'Spiller ' + spiller + ' tabte (forsøgte at tage ' + antal + ' tændstikker)';
        farve = '#ffcccc'; // rød baggrund
        tilbage = 0;
    } else {
        tilbage = tilbage - antal; // tag tændstikker!
        status = 'Spiller ' + spiller + ' tog ' + antal + ' fra bordet (' + tilbage + ' tilbage)';
        if ( tilbage == 0 ) { // vundet!
            status = 'Spiller ' + spiller + ' vandt spillet!';
            farve = '#ccffcc'; // grøn baggrund
        } else {
            spiller = 3 - spiller; // skift spiller (1 <-> 2)
            if ( spiller == 1 ) farve = 'white'; // hvid baggrund
            else farve = '#ffffcc'; // gul baggrund
        }
    }
    /* -- gør klar til næste runde -- */
    gentegnBordet();
    document.getElementById('status').innerHTML += '<li>' + status + '</li>'; // opdater status
    document.body.style.backgroundColor = farve; // opdater baggrundsfarve
    document.getElementById('spiller').innerHTML = spiller; // opdater spiller
}
```

# Lav din egen Burger



<http://itu.dk/people/brabrand/burger/burger.html>

# World Cup 2018

```
<html><body><script>  
var x = new XMLHttpRequest();  
x.open('GET', "https://raw.githubusercontent.com/openfootball/world-cup.json/master/2018/worldcup.json", false);  
x.send();  
var json = x.responseText; ← http  
var resultat = JSON.parse(json);  
document.write('<h3>' + resultat.name + '</h3>');  
  
document.write('<code>');  
for (let round of resultat.rounds) { // iterate list  
    for (let match of round.matches) { // iterate list  
        document.write(match.team1.code + ' - ' + match.team2.code + ': ');  
        document.write(match.score1 + ' - ' + match.score2 + '<br/>');  
    }  
}  
document.write('</code>');  
</script></body></html>
```

JSON



## World Cup 2018

RUS	-	KSA	:	5	-	0
EGY	-	URU	:	0	-	1
POR	-	ESP	:	3	-	3
MAR	-	IRN	:	0	-	1
FRA	-	AUS	:	2	-	1
PER	-	DEN	:	0	-	1
	:					
FRA	-	CRO	:	4	-	2

<https://www.w3schools.com/code/tryit.asp?filename=FSKFZ1JAYMDA>

# World Cup 2018 (★)

Modificér så den *Danmarks kampe* printes:

```
<html><body><script>  
var x = new XMLHttpRequest();  
x.open('GET', "https://raw.githubusercontent.com/openfootball/world-cup.json/master/2018/worldcup.json", false);  
x.send();  
var json = x.responseText;  
var resultat = JSON.parse(json); ← http  
document.write('<h3>' + resultat.name + '</h3>');      JSON  
  
document.write('<code>');  
for (let round of resultat.rounds) { // iterate list  
    for (let match of round.matches) { // iterate list  
        document.write(match.team1.code + ' - ' + match.team2.code + ': ');  
        document.write(match.score1 + ' - ' + match.score2 + '<br/>');  
    }  
}  
document.write('</code>');  
</script></body></html>
```



## World Cup 2018

PER	-	DEN	:	0	-	1
DEN	-	AUS	:	1	-	1
DEN	-	FRA	:	0	-	0
CRO	-	DEN	:	1	-	1



# OPGAVE



# World Cup 2018 (★)

```
<html><body><script>  
var x = new XMLHttpRequest();  
x.open('GET', "https://raw.githubusercontent.com/openfootball/world-cup.json/master/2018/worldcup.json", false);  
x.send();  
var json = x.responseText;  
var resultat = JSON.parse(json); ← JSON  
document.write('<h3>' + resultat.name + '</h3>');  
  
document.write('<code>');  
for (let round of resultat.rounds) { // iterate list  
    for (let match of round.matches) { // iterate list  
        if (match.team1.code=='DEN' || match.team2.code=='DEN') {  
            document.write(match.team1.code + ' - ' + match.team2.code + ': ');  
            document.write(match.score1 + ' - ' + match.score2 + '<br/>');  
        }  
    }  
}  
document.write('</code>');  
</script></body></html>
```

http



JSON

## World Cup 2018

PER	-	DEN	:	0	-	1
DEN	-	AUS	:	1	-	1
DEN	-	FRA	:	0	-	0
CRO	-	DEN	:	1	-	1

<https://www.w3schools.com/code/tryit.asp?filename=G49UTW1OGGAS>