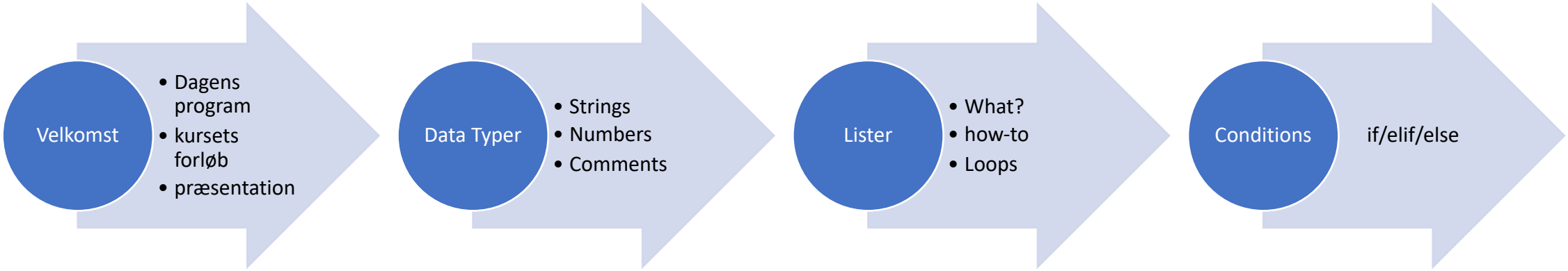
EVU Python

Dagens program



Program

- Tidsplan

9:00	9:15	9:30 - P	10:00	10:15	10:30 - P	11:15	11:40	12:00
Velkomst	Intro	Præsens	Variabler	Variabler	GØ	Lister	Lister	PAUSE
				Øvelser			Øvelser	
12:30	12:45	13:00 - P	13:15	13:30	13:45 - P	14:15	14:30	-
GØ	Conditio	Conditio	Conditio	GØ	Strings	Strings	GØ & Recap	
			Øvelser			Øvelser		

Velkomst

kursets forløb

27/1

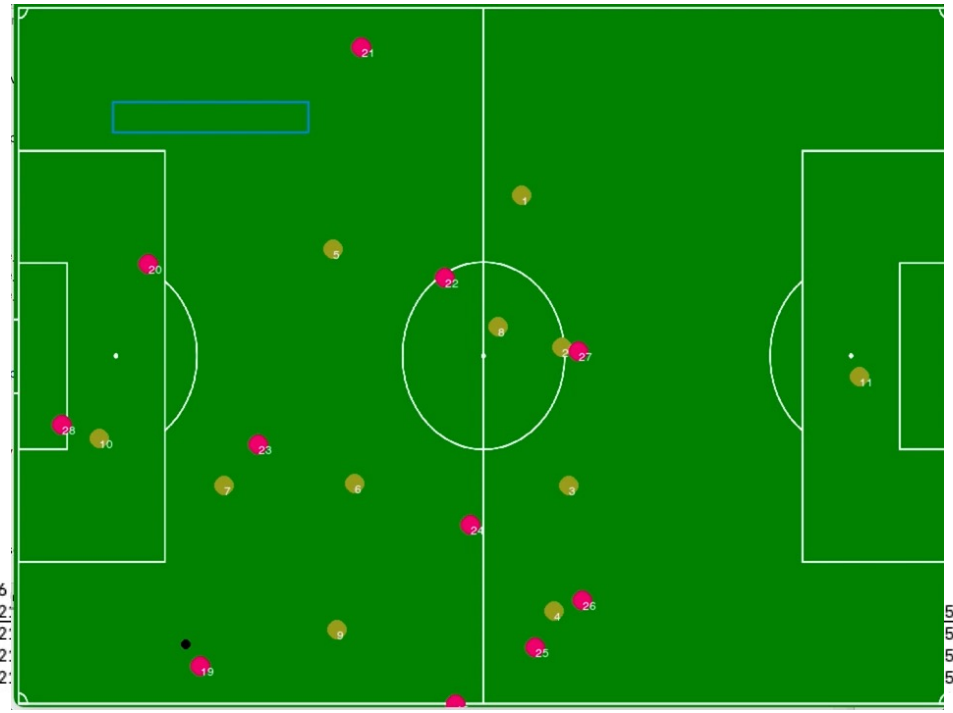
10/2

24/2

10/3

Teaser

Sports Analytics

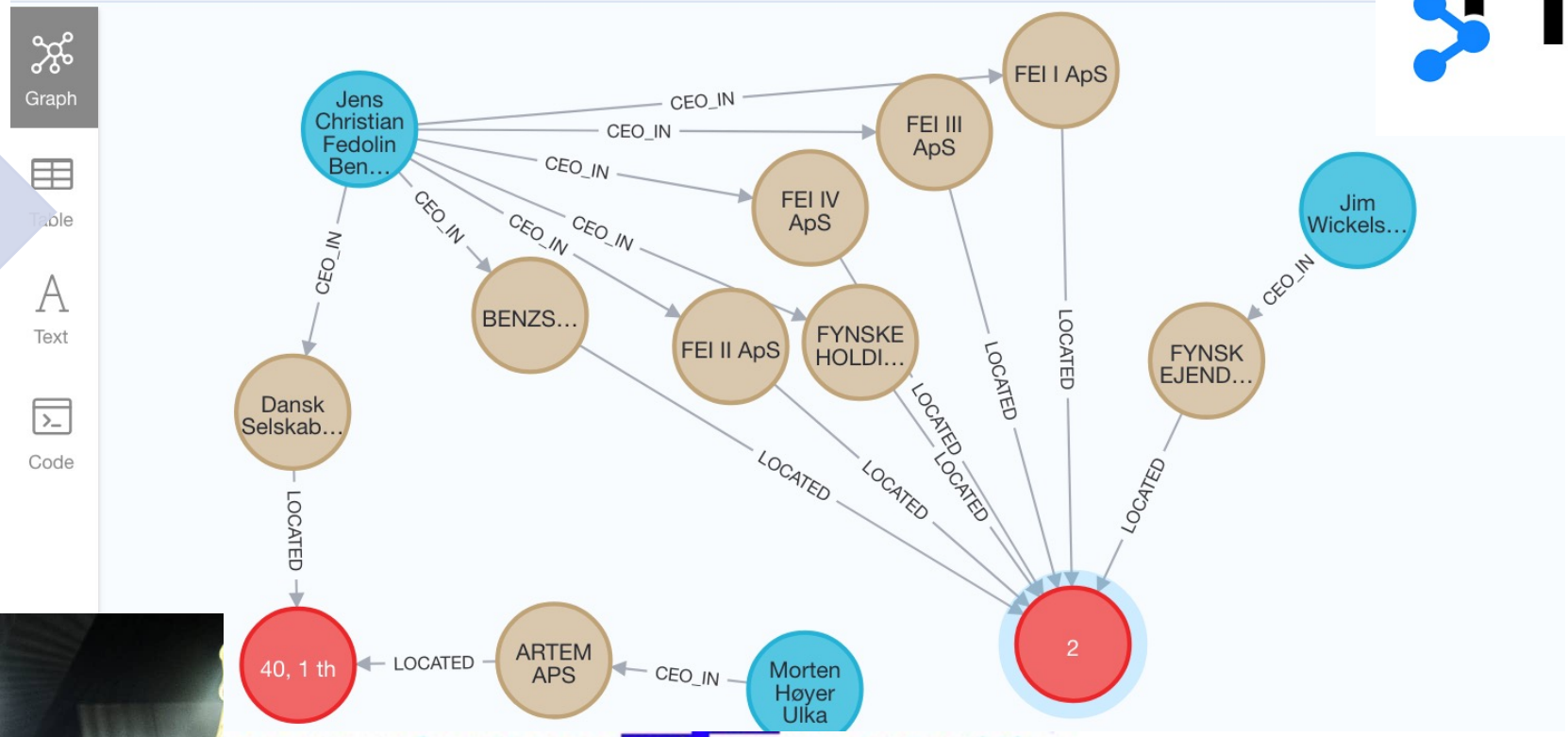


```
Period,Frame,Time [s],Player11,,Player1,,Player2,,Player3,,Player4,,Player5,,Player6,
1,1,0.04,0.00082,0.48238,0.32648,0.65322,0.33701,0.48863,0.30927,0.35529,0.32137,0.2 5243,0.43269,NaN,NaN,NaN,NaN,NaN,NaN,0.45472,0.38709
1,2,0.08,0.00096,0.48238,0.32648,0.65322,0.33701,0.48863,0.30927,0.35529,0.32137,0.2 5243,0.43269,NaN,NaN,NaN,NaN,NaN,NaN,0.49645,0.40656
1,3,0.12,0.00114,0.48238,0.32648,0.65322,0.33701,0.48863,0.30927,0.35529,0.32137,0.2 5243,0.43269,NaN,NaN,NaN,NaN,NaN,NaN,0.53716,0.42556
1,4,0.16,0.00121,0.48238,0.32622,0.65317,0.33687,0.48988,0.30944,0.35554,0.32142,0.2 55236,0.43313,NaN,NaN,NaN,NaN,NaN,NaN,0.55346,0.42231
```

Teaser

Fraud Detection

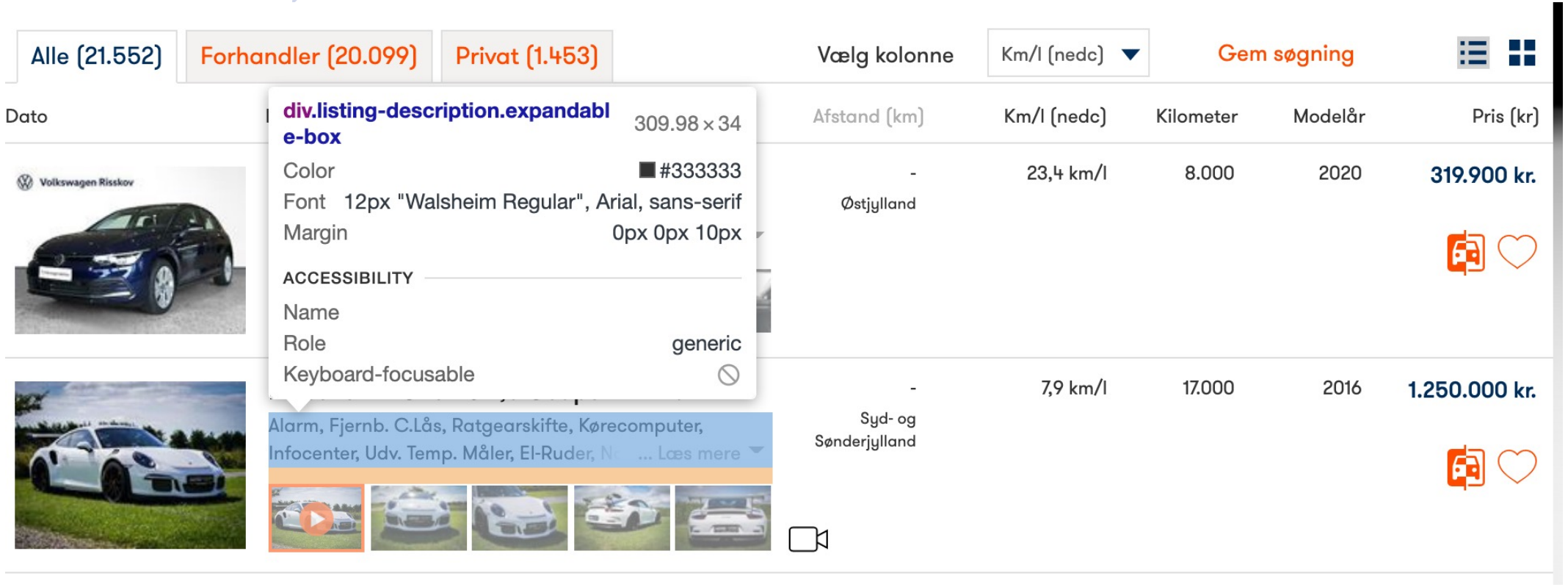
```
neo4j$ match (n:Location) where id(n)=55 return n
```



```
c174">9700</c:AddressOfAuditorPostCodeIdentifier>
Brønderslev</c:AddressOfAuditorDistrictName>
rdjyske Bank</c:NameOfFinancialInstitution>
Vodskovvej</c:AddressOfFinancialStreetName>
extRef="c174">43</c:AddressOfFinancialStreetBuildingIdentifier>
="c174">9310</c:AddressOfFinancialPostCodeIdentifier>
">Vodskov</c:AddressOfFinancialDistrictName>
ester Hassing</e:PlaceOfSignatureOfStatement>
2017-11-07</e:DateOfApprovalOfAnnualReport>
ef="c637">Poul Erik Madsen</d:NameAndSurnameOfMemberOfExecutiveBoard>
tRef="c646">Poul Erik Madsen</d:NameAndSurnameOfMemberOfSupervisoryBoard>
tRef="c647">Martin Nørgaard</d:NameAndSurnameOfMemberOfSupervisoryBoard>
<f:SignatureOfAuditorsPlace contextRef="c174">Brønderslev</f:SignatureOfAuditorsPlace>
```

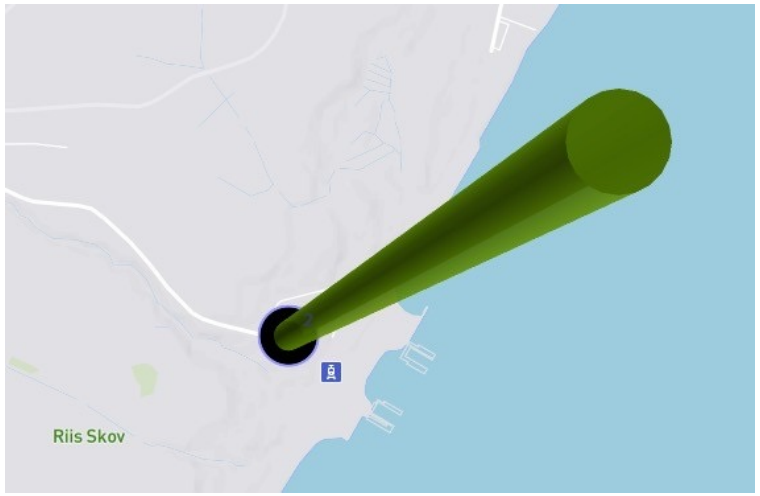
Teaser

Web Scraping



Teaser

REST API



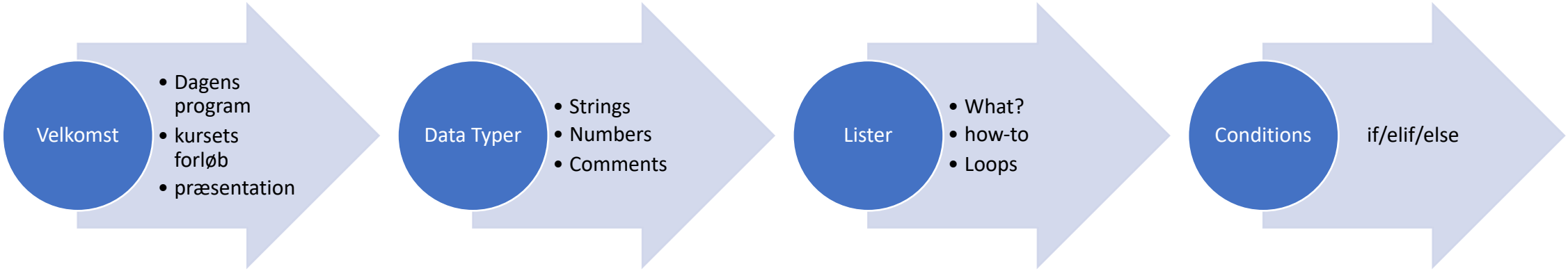
http shell javascript

```http
GET /devices HTTP/1.1
Authorization: Bearer {BEARER_TOKEN}
Host: api.cityflow.live
```

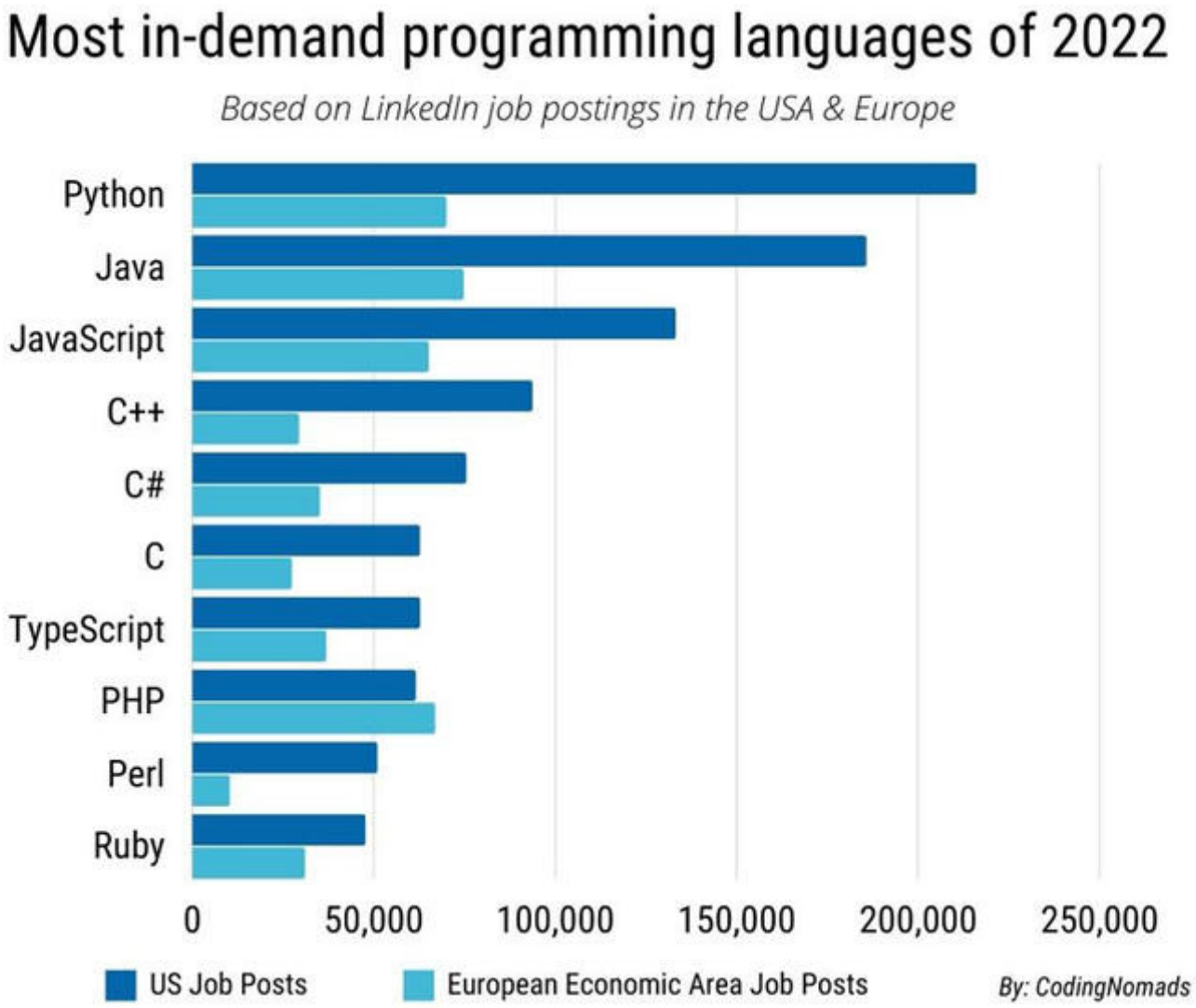

```
},
{
  id: e00fce689f02a96799f34fc2,
  type: 150,
  location: 190,
  latitude: 56.1770897,
  longitude: 10.2296247,
  location_name: Salonvejen,
  city: Risskov,
  country: Denmark,
  roles: [
    4
  ],
  permissions: [],
  tags: [
    Risskov
  ]
},
```

Pause

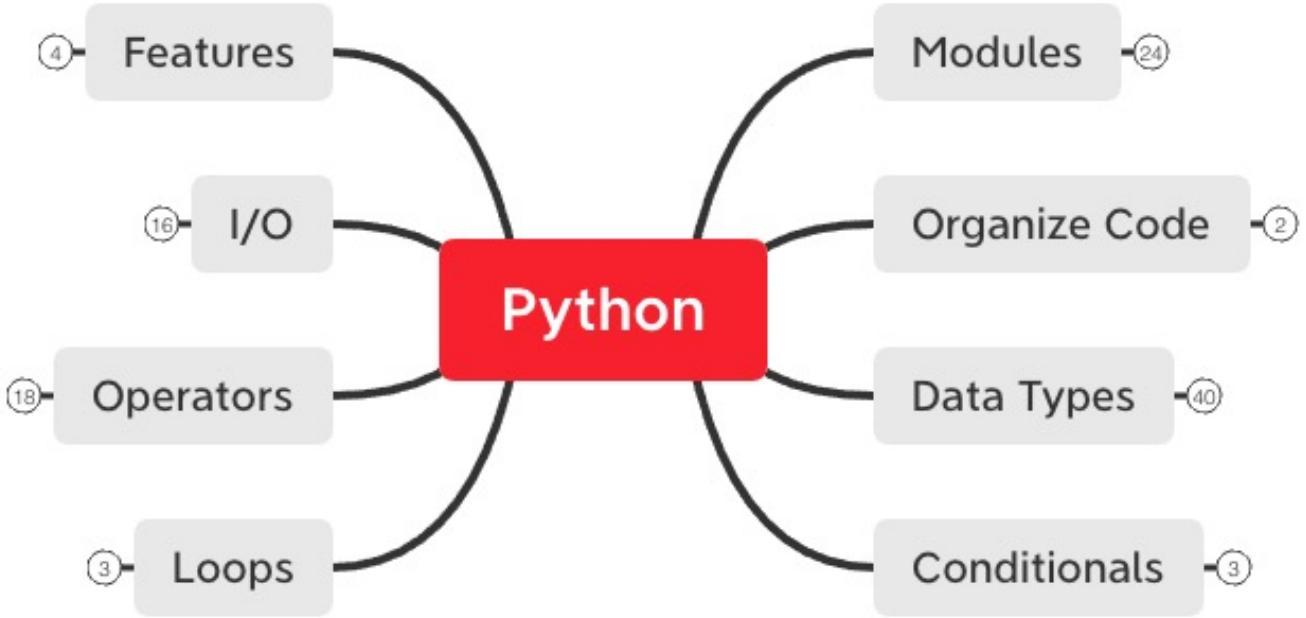
Dagens program



Python Stats



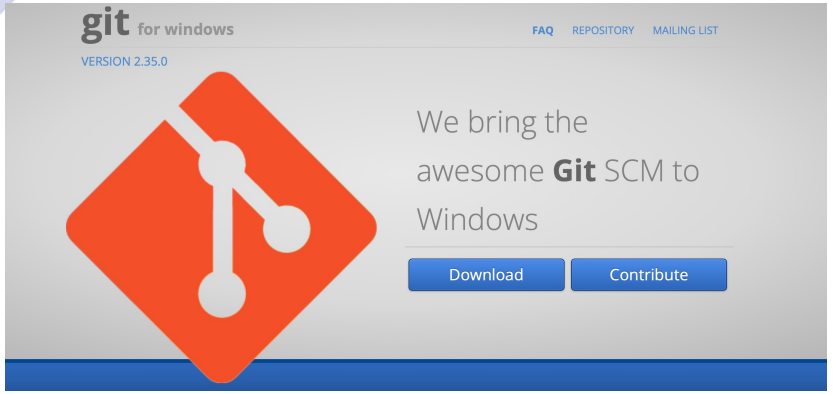
Python Struktur

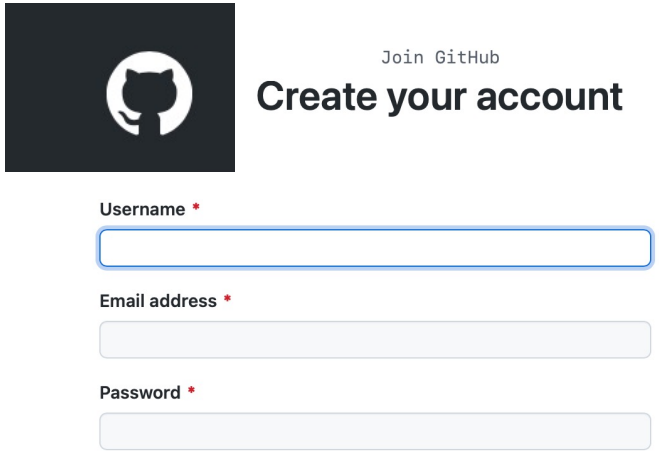


Python

Tools

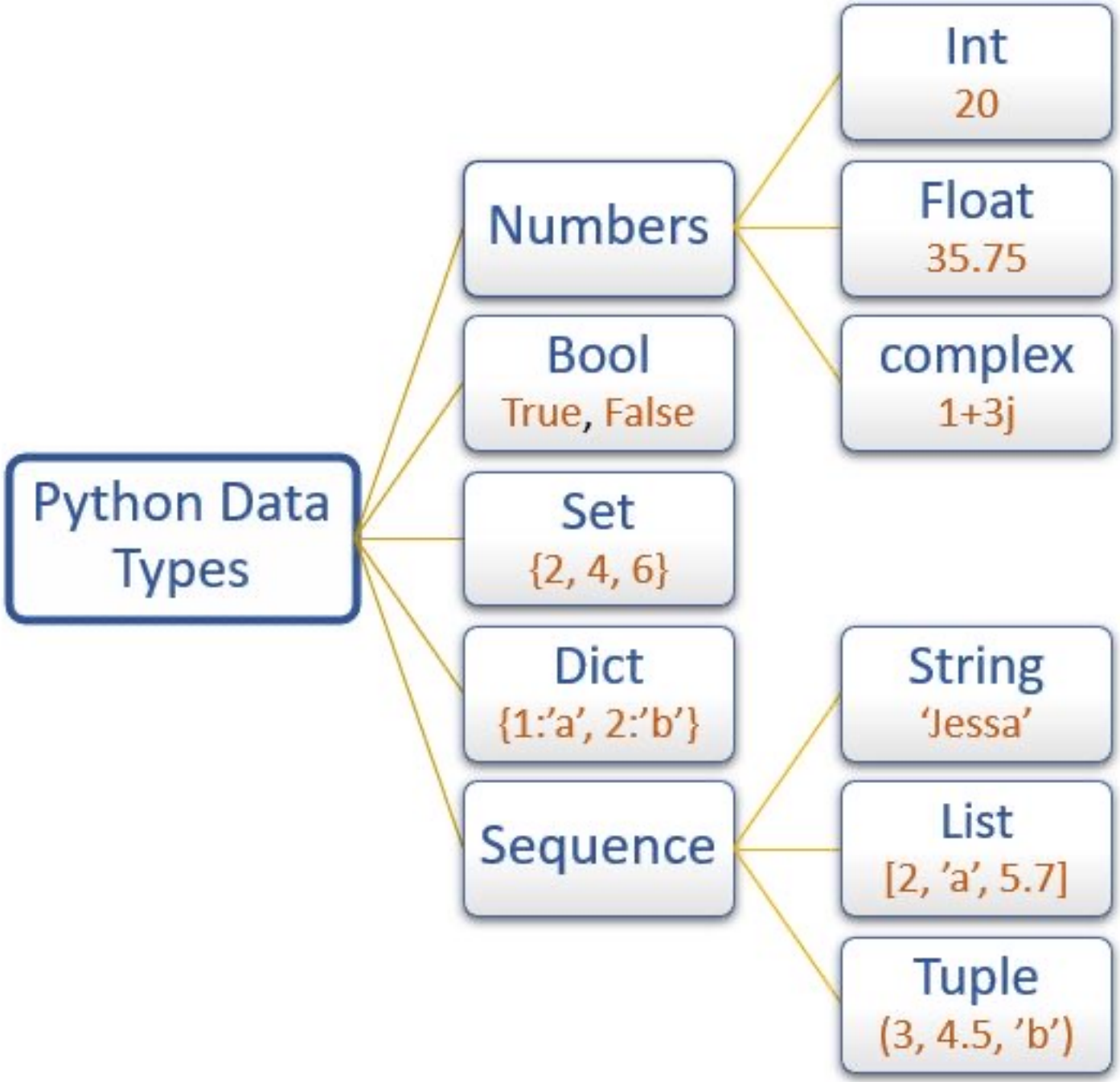
- Install GitBash (windows) or Git
- Create Git-account





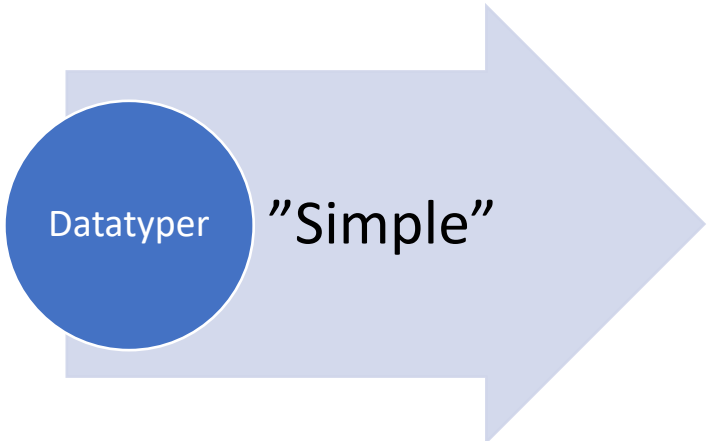
Datatyper

”Simple”

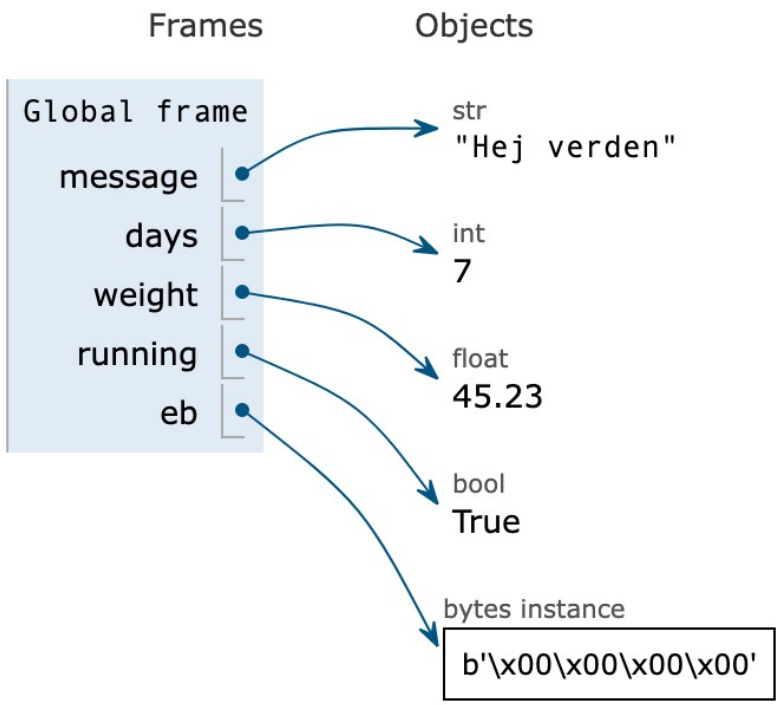


A Python variable is a reserved memory location to store values.

Every value in Python has a datatype.



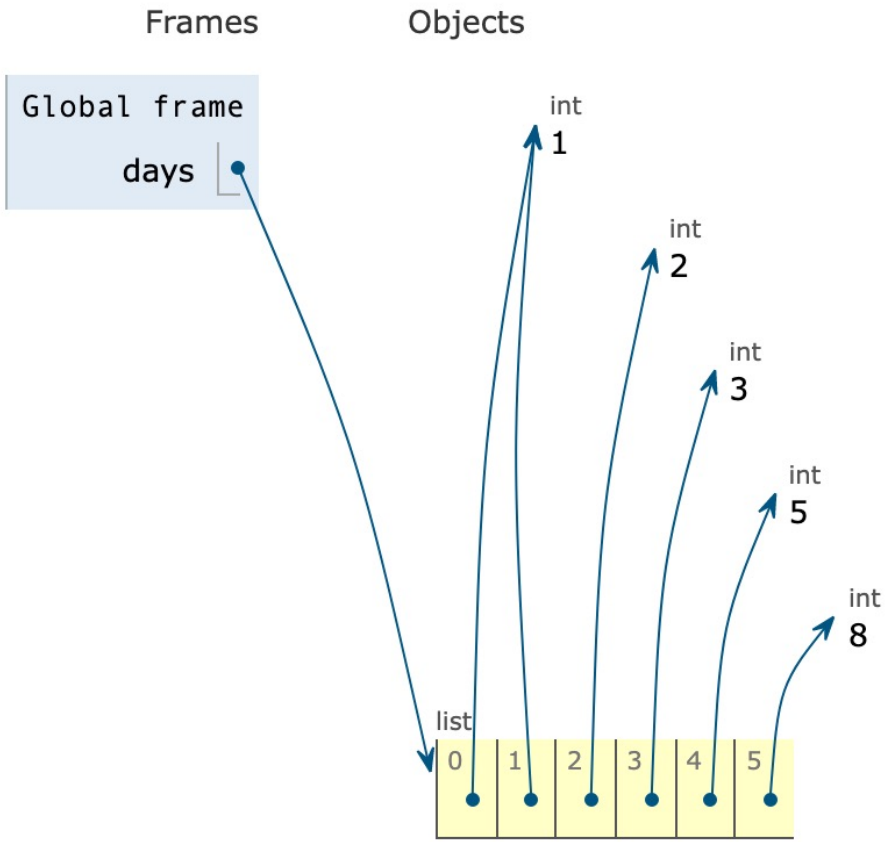
A Python variable is a reserved memory location to store values.
Every value in Python has a datatype.



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

Datatyper

Komplekse



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`

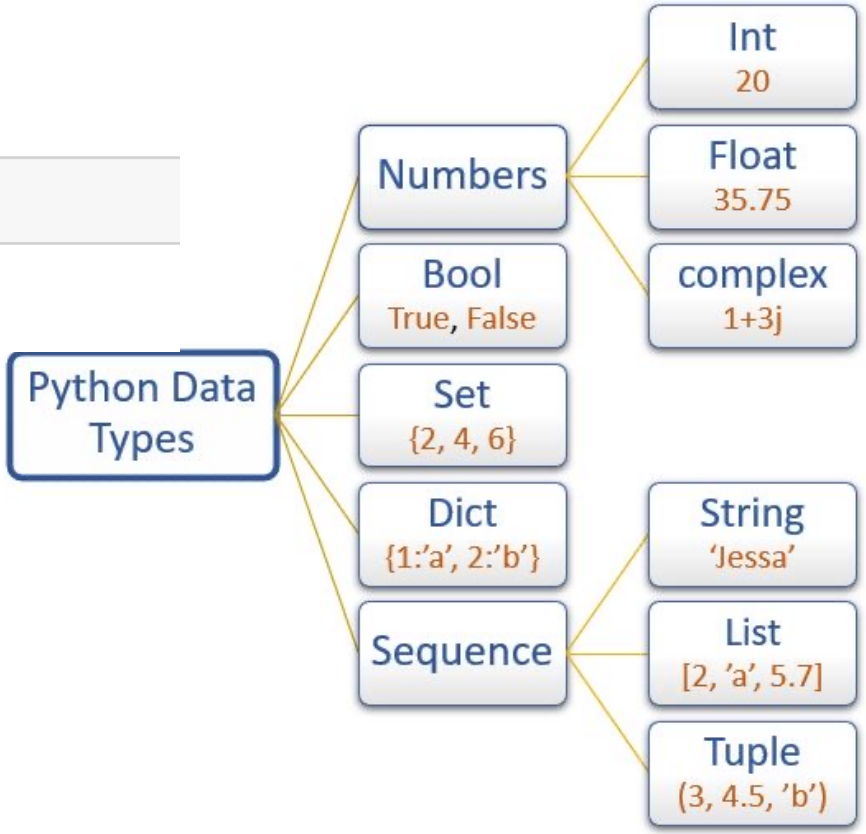
Variabler

How to?

Dynamic Typing

Python uses *dynamic typing*, meaning you can reassign variables to different data types. This makes Python very flexible in assigning data types; it differs from other languages that are *statically typed*.

```
In [7]: type("Jessa")
Out[7]: str
```



Øvelse

```
In [8]: type( )
Out[8]: int
```

Variabler

Avoid Keywords

Python Keywords

Each of the following keywords has a specific meaning, and you'll see an error if you try to use them as a variable name.

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Variabler

Builtin functions

Python Built-in Functions

abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	apply()
delattr()	help()	next()	setattr()	buffer()
dict()	hex()	object()	slice()	coerce()
dir()	id()	oct()	sorted()	intern()

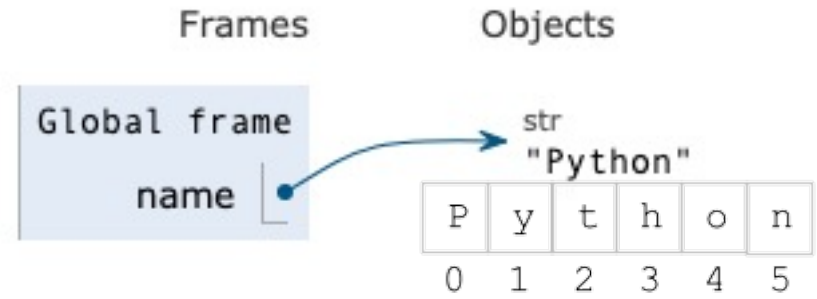
Variabler

Strings

Definition

A string is simply a series of characters (” ” or ‘ ‘)
”*sequence of bytes representing unicode characters*”

Memory



Standard

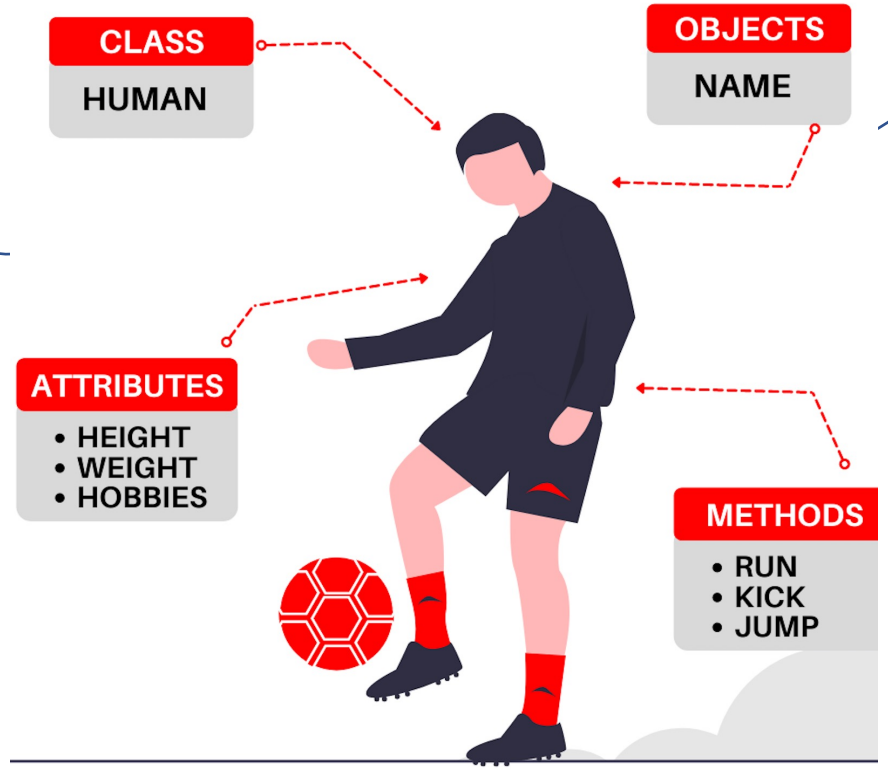
”Kurt Verner”
” 1234”

Special

”Kurt \u0023Tegne”
"Kurt\"Med\"Tegn"

Variabler Strings

The Dot-notation




```
messi.run(12)
messi.jump()
....

”messi”.toupper()
”messi”.strip()
```

Variabler

Strings

Common built-in functions

print(”kurt Verner”)

len(”kurt Verner”)

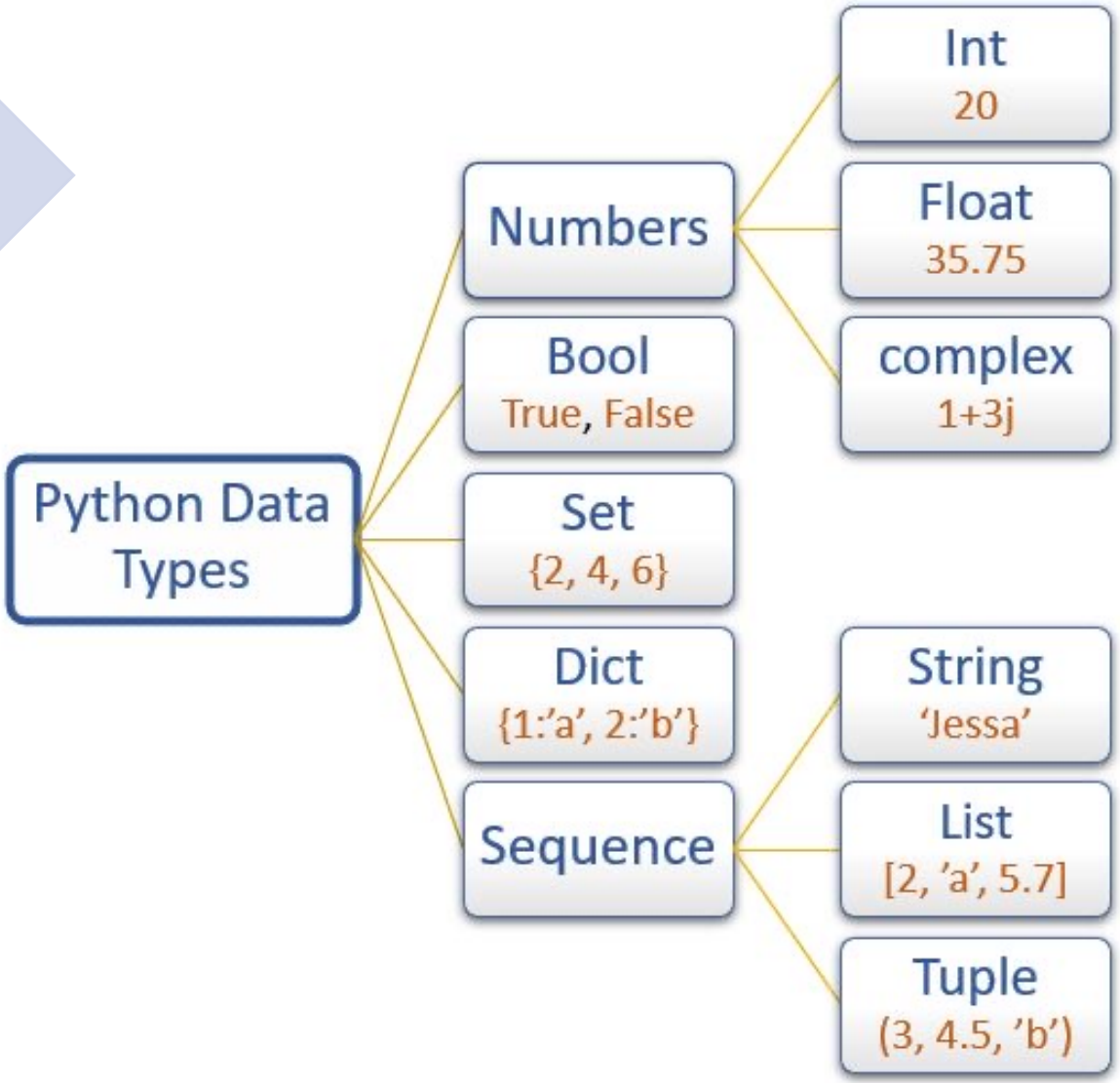
Common methods

Function	Description
format()	It's used to create a formatted string from the template string and the supplied values.
split()	Python string split() function is used to split a string into the list of strings based on a delimiter.
join()	This function returns a new string that is the concatenation of the strings in iterable with string object as a delimiter.
strip()	Used to trim whitespaces from the string object.
format_map()	Python string format_map() function returns a formatted version of the string using substitutions from the mapping provided.
upper()	We can convert a string to uppercase in Python using str.upper() function.
lower()	This function creates a new string in lowercase.
replace()	Python string replace() function is used to create a new string by replacing some parts of another string.
find()	Python String find() method is used to find the index of a substring in a string.
translate()	Python String translate() function returns a new string with each character in the string replaced using the given translation table.

Pause

Variabler

Numbers



Data

Operators

Precedence	Associativity	Operator	Description
18	Left-to-right	()	Parentheses (grouping)
17	Left-to-right	*f*(args...)	Function call
16	Left-to-right	*x*[index:index]	Slicing
15	Left-to-right	*x*[index]	Array Subscription
14	Right-to-left	**	Exponentiation
13	Left-to-right	~*x*	Bitwise not
12	Left-to-right	+*x* -*x*	Positive, Negative
11	Left-to-right	* / %	Multiplication Division Modulo
10	Left-to-right	+ -	Addition Subtraction
9	Left-to-right	<< >>	Bitwise left shift Bitwise right shift
8	Left-to-right	&	Bitwise AND
7	Left-to-right	^	Bitwise XOR
6	Left-to-right	|	Bitwise OR
5	Left-to-right	in, not in, is, is not, <, <=, >, >=, <>, == !=	Membership Relational Equality Inequality
4	Left-to-right	not *x*	Boolean NOT
3	Left-to-right	and	Boolean AND
2	Left-to-right	or	Boolean OR
1	Left-to-right	lambda	Lambda expression

Variabler

Numbers

- Int 20
- Float 35.75
- complex 1+3j

11	Left-to-right	* / %	Multiplication Division Modulo
10	Left-to-right	+ -	Addition Subtraction

Functions

abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	apply()
delattr()	help()	next()	setattr()	buffer()
dict()	hex()	object()	slice()	coerce()
dir()	id()	oct()	sorted()	intern()

Datatyper

Øvelser

- 2-3 – Personal message
- 2-4 – Name – lower,upper
- 2-5 – Famous quote – en rigtig kilde
- 2-7 - Strange name (strip)
- 2-8 – 8-tals stykkker (add,sub,mult og div)

Numbers

13 + 5 * 2 - 4 skal give 32

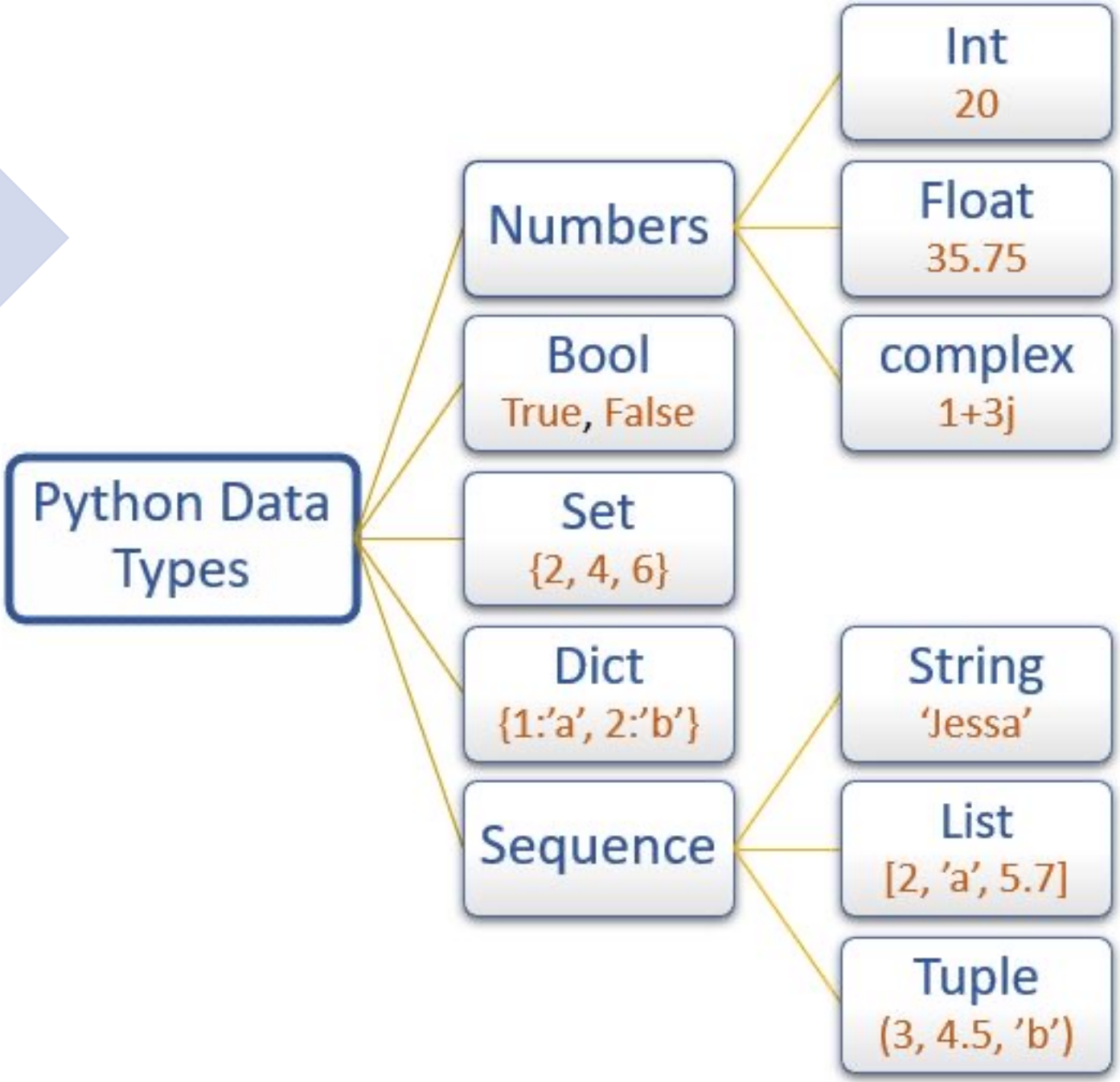
```
In [ ]:
```

13 + 5 * 2 - 4 / 2 - 13 skal give 3.0

Pause

Datatyper

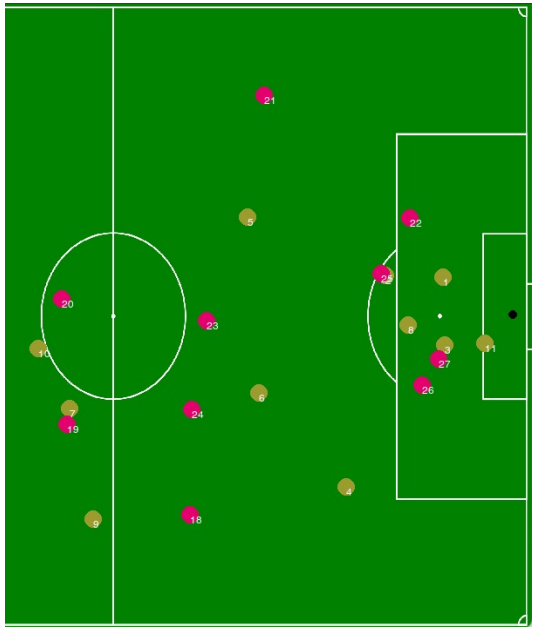
Sekventielle

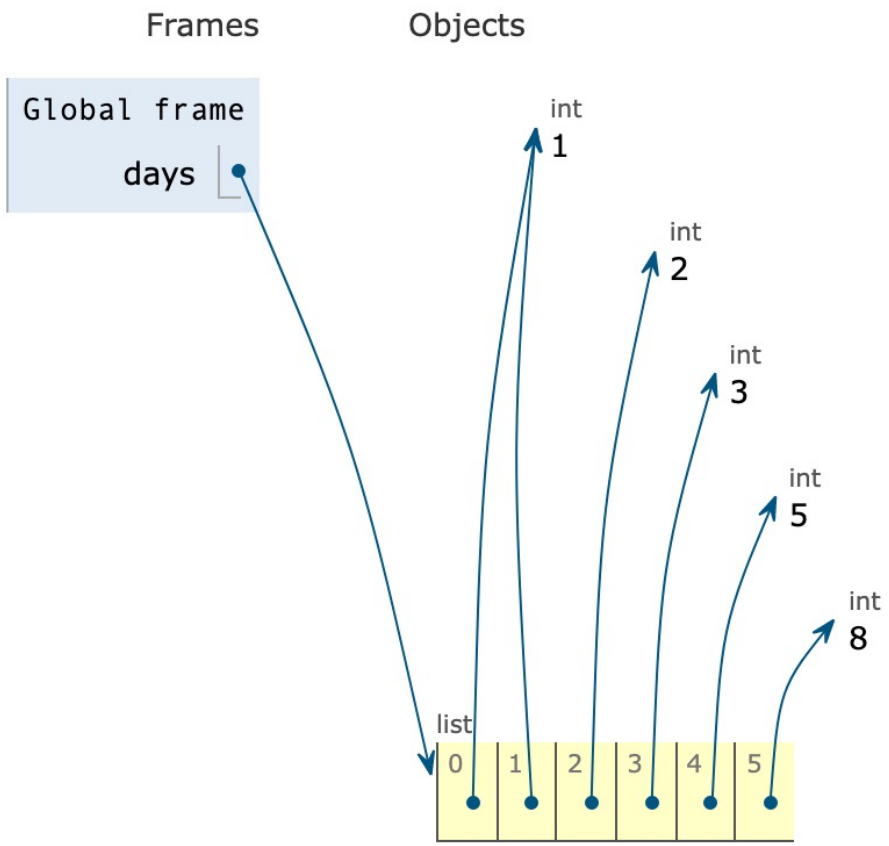


Datatyper

Lister – hvad?

playingPlayers = {list: 22}
- 00 = {str} 'player1_y'
- 01 = {str} 'player2_y'
- 02 = {str} 'player3_y'
- 03 = {str} 'player4_y'
- 04 = {str} 'player5_y'
- 05 = {str} 'player6_y'
- 06 = {str} 'player7_y'
- 07 = {str} 'player8_y'





Datatyper

Lister - Metoder

Method	Description	Example

`lst.append(x)`	Appends element `x` to the list `lst`.	`>>> l = []`
		`>>> l.append(42)`
		`>>> l.append(21)`
		`[42, 21]`

`lst.clear()`	Removes all elements from the list lst–which becomes empty.	`>>> lst = [1, 2, 3, 4, 5]`
		`>>> lst.clear()`
		`[]`

`lst.copy()`	Returns a copy of the list `lst`. Copies only the list, not the elements in the list (shallow copy).	`>>> lst = [1, 2, 3]`
		`>>> lst.copy()`
		`[1, 2, 3]`

`lst.count(x)`	Counts the number of occurrences of element `x` in the list `lst`.	`>>> lst = [1, 2, 42, 2, 1, 42, 42]`
		`>>> lst.count(42)`
		`3`
		`>>> lst.count(2)`
		`2`

`lst.extend(iter)`	Adds all elements of an iterable `iter` (e.g. another list) to the list `lst`.	`>>> lst = [1, 2, 3]`
		`>>> lst.extend([4, 5, 6])`
		`[1, 2, 3, 4, 5, 6]`

`lst.index(x)`	Returns the position (index) of the first occurrence of value `x` in the list `lst`.	`>>> lst = ["Alice", 42, "Bob", 99]`
		`>>> lst.index("Alice")`
		`0`
		`>>> lst.index(99, 1, 3)`
		`ValueError: 99 is not in list`

`lst.insert(i, x)`	Inserts element `x` at position (index) `i` in the list `lst`.	`>>> lst = [1, 2, 3, 4]`
		`>>> lst.insert(3, 99)`
		`[1, 2, 3, 99, 4]`

`lst.pop()`	Removes and returns the final element of the list `lst`.	`>>> lst = [1, 2, 3]`
		`>>> lst.pop()`
		`3`
		`>>> lst`
		`[1, 2]`

`lst.remove(x)`	Removes and returns the first occurrence of element `x` in the list `lst`.	`>>> lst = [1, 2, 99, 4, 99]`
		`>>> lst.remove(99)`
		`>>> lst`
		`[1, 2, 4, 99]`

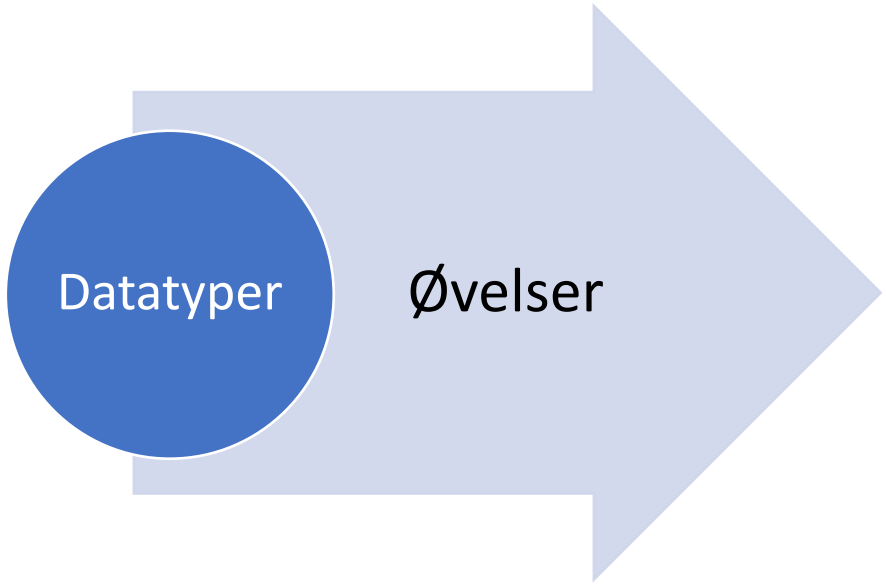
`lst.reverse()`	Reverses the order of elements in the list `lst`.	`>>> lst = [1, 2, 3, 4]`
		`>>> lst.reverse()`
		`>>> lst`
		`[4, 3, 2, 1]`

`lst.sort()`	Sorts the elements in the list `lst` in ascending order.	`>>> lst = [88, 12, 42, 11, 2]`
		`>>> lst.sort()`
		`# [2, 11, 12, 42, 88]`
		`>>> lst.sort(key=lambda x: str(x)[0])`
		`# [11, 12, 2, 42, 88]`

Datatyper

Lister - Funktioner

abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	apply()
delattr()	help()	next()	setattr()	buffer()
dict()	hex()	object()	slice()	coerce()
dir()	id()	oct()	sorted()	intern()



- 3-1 - Venneliste
- 3-2 - Brug listen til velkomst besked
- 3-4 - Gæsteliste
- 3-5 – Ændringer
  - replace
- 3-8 – TopSightsList
  - Print alfabetisk
  - Reverse

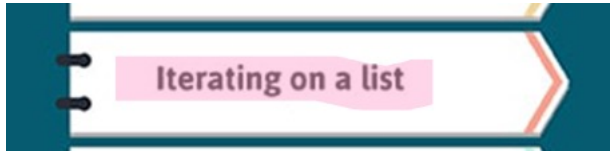
Pause

Datatyper

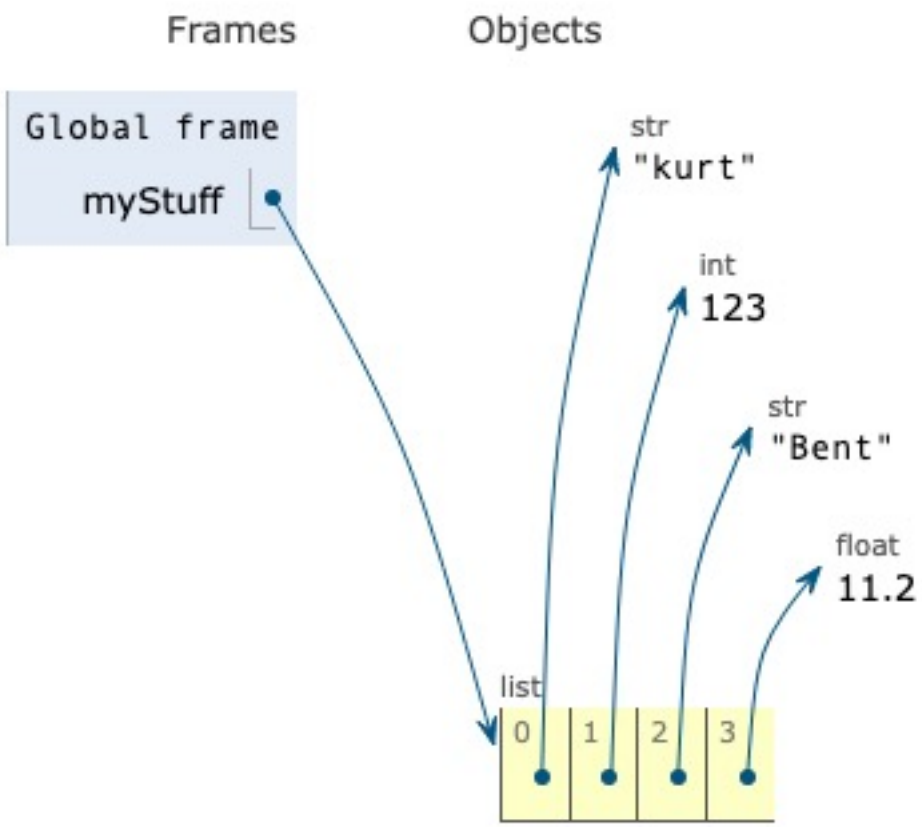
Lister in Action

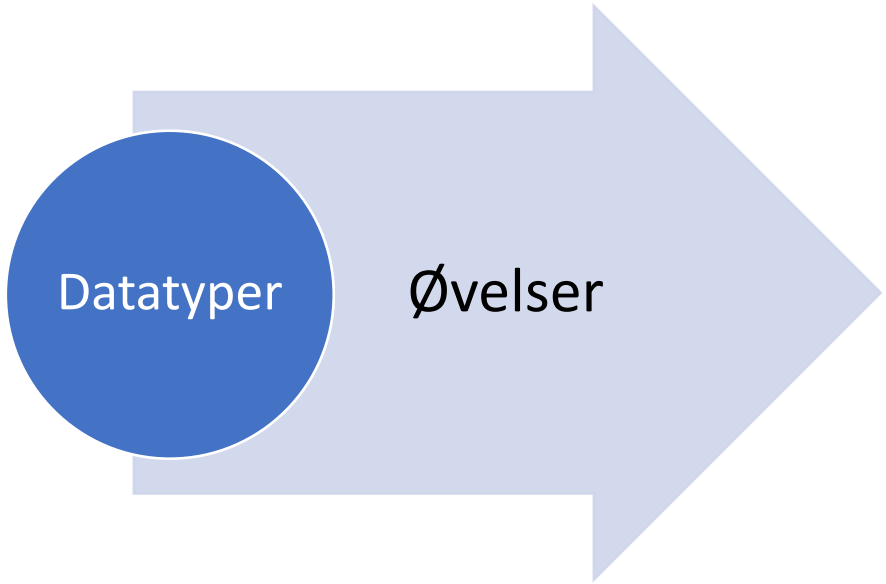
Datatyper

Lister in Action

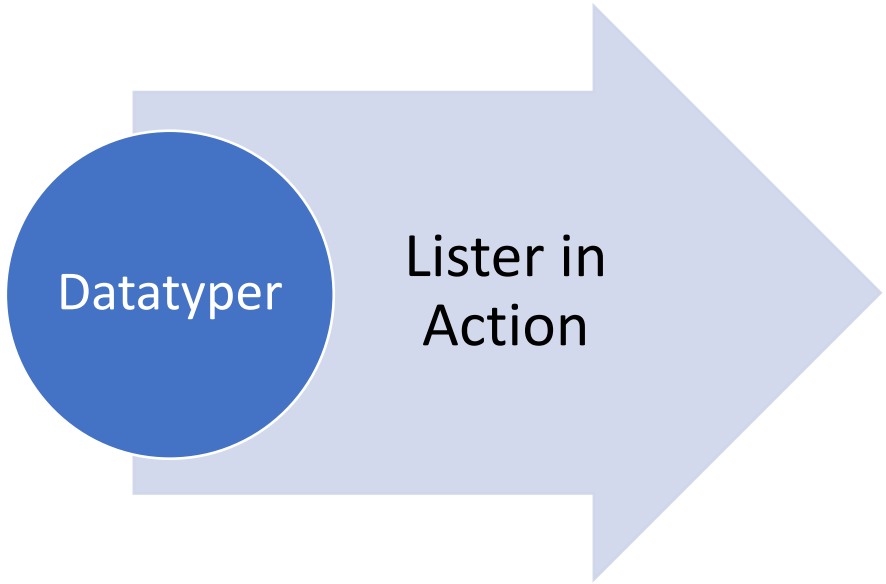



```
1  myStuff = ["kurt",123,"Bent",11.2]
2  for item in myStuff:
3      print(item)
4
```





- 4-2 – Dyr med fællestræk
  - Loop
  - konklusion
- 4-3 – Tæl til 20
- 4-6 – Tæl ulige tal
  - range
  - Modulo
  - List comprehension
- 4-10 Slicing pizzas
  - First 3
  - Middle
  - Last
- 4-11 new pizza from copy
  - Add one to both



Lister - Biler

In [2]:
```
cars=[]
fh=open('/Users/thor/Git/EVUF22LES1/cars.csv','r')
lines=fh.readlines()
```

Pause

Data

Operators

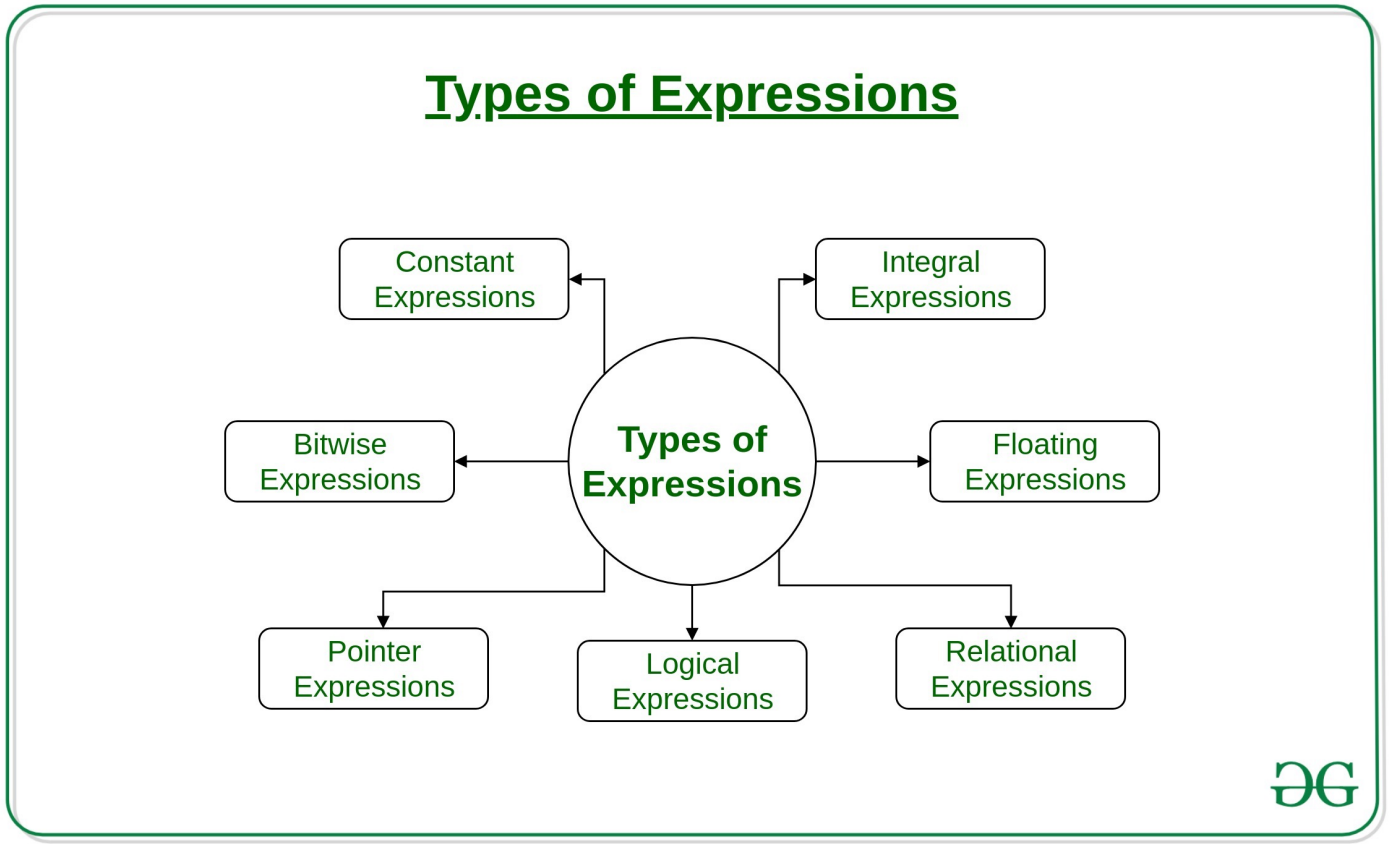
Precedence	Associativity	Operator	Description
18	Left-to-right	()	Parentheses (grouping)
17	Left-to-right	*f*(args...)	Function call
16	Left-to-right	*x*[index:index]	Slicing
15	Left-to-right	*x*[index]	Array Subscription
14	Right-to-left	**	Exponentiation
13	Left-to-right	~*x*	Bitwise not
12	Left-to-right	+*x* -*x*	Positive, Negative
11	Left-to-right	* / %	Multiplication Division Modulo
10	Left-to-right	+ -	Addition Subtraction
9	Left-to-right	<< >>	Bitwise left shift Bitwise right shift
8	Left-to-right	&	Bitwise AND
7	Left-to-right	^	Bitwise XOR
6	Left-to-right	|	Bitwise OR
5	Left-to-right	in, not in, is, is not, <, <=, >, >=, <>, == !=	Membership Relational Equality Inequality
4	Left-to-right	not *x*	Boolean NOT
3	Left-to-right	and	Boolean AND
2	Left-to-right	or	Boolean OR
1	Left-to-right	lambda	Lambda expression

Variabler

Statement

```
print("The answer is: " + str(round(x * 5, 1)) )
```

Expression

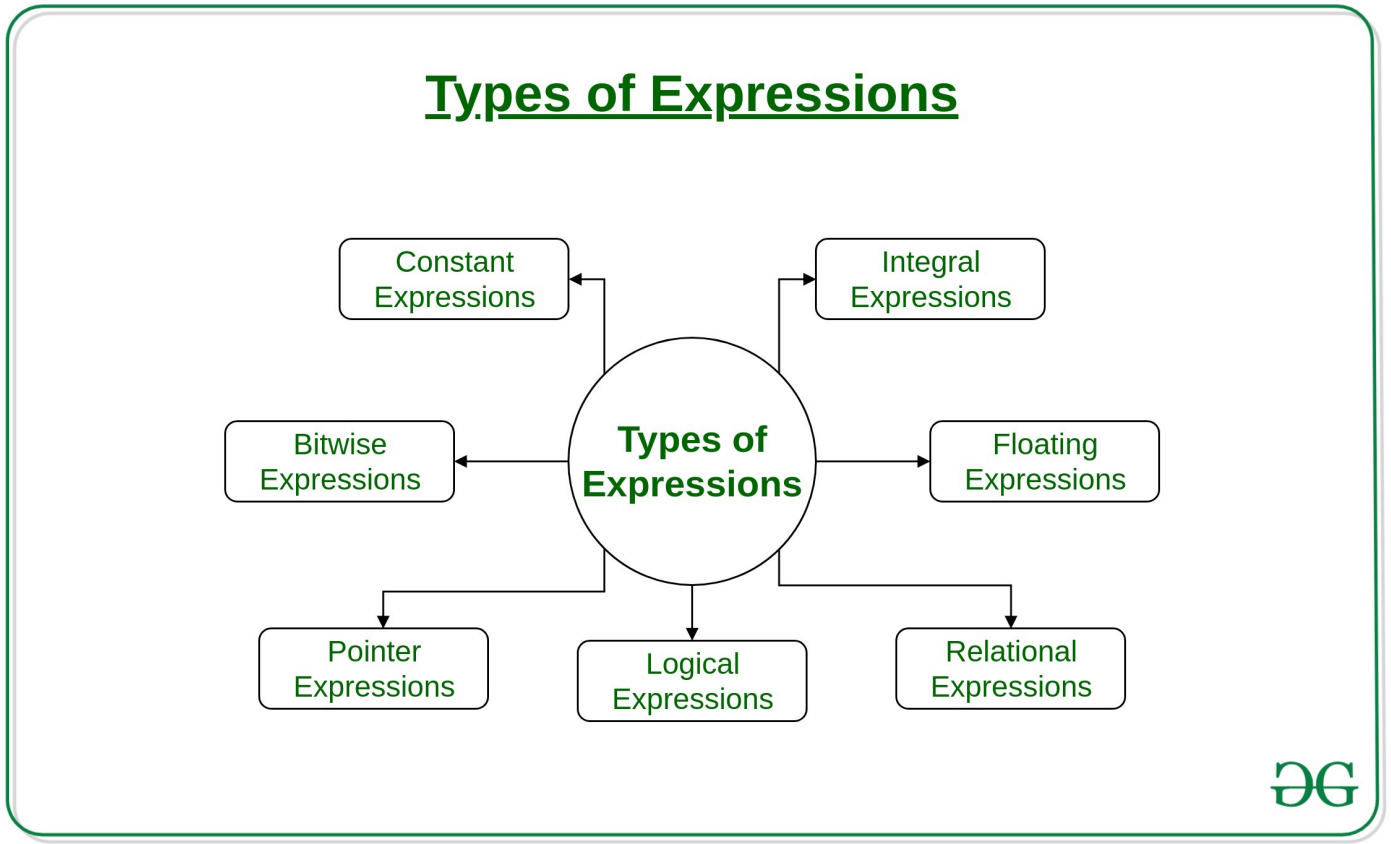


Variabler

Statement

```
print("The answer is: " + str(round(x * 5, 1)) )
```

Expression



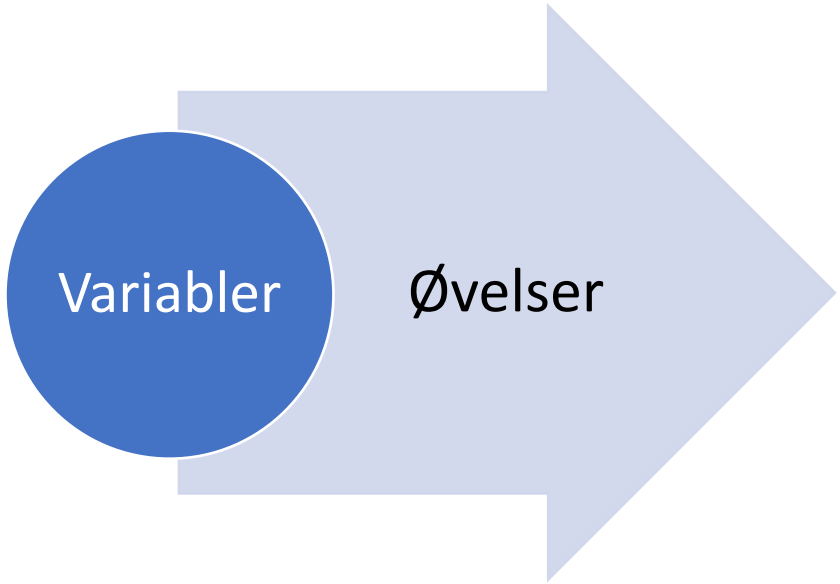
Variabler

Booleans

Bool
True, False

Operators

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
!=	Not equal to	5!=2	True



- 5-5 – Alien colors

Lister og conditions

Booleans

res=not(7<1) or (7 > 4) kan den blive til falsk vha parentes?

```
In [ ]:
1
```

Conditions og biler

Lav en liste og put alle BMW'er i listen