
Deklaration av variabler, input och output: variabler måste deklareras med en typ innan de används. Input från användaren tas med funktionen get_input där argumentet blir meddelandet till användaren. Print funktionen skriver ut argumentet på en ny rad.

```
int x = 0;
string b;
b += get_input("Skriv in ett något:");
print(b);

output:
==>"Skriv in ett något:"
2
==>2
```

Arrayer och operatorer: Vårt språk föredrar användning av operatorer över användning av inbyggda funktioner. Detta gör det lättare för programmeraren att komma ihåg de olika uttrycken. Arrayer använder += operatorn för att appenda ett givet värde lägst bak i arrayen, -= används antagligen för att ta bort alla entiteter med ett givet värde eller för att ta bort ett särskilt index.

(Array/Vector? Fråga på seminariet.)

```
array a = [1,3,3];
a[1] = 2;
a += 4;
print(a);
output:
[1,2,3,4]

a -= 2;
print(a);
output:
[1,3,4]

a -= [2];
print (a);
output:
[1,3]

a[2] -=3;
output:
[1,0]
```

Aritmetiska operatorer: Atrimetiska operatorer följer de matematiska beräkningsreglerna

```
print( 1-2+5 );
print( 1-(2+5) );
print( 2*5+2 );
print( 2*(5+2) );

output:
4
6
12
14
```

Starkt typat: vårt språk gör ingen inbyggd omvandling för datatyper. Detta gör det svårare att råka göra misstag vilket är bra för nybörjare.

```
string b = "12";
int a = 8;
print (a+b);
print (int(b)+a);

output:
TypeError
20
```

Definition av funktioner: funktioner defineras med orden ”def [returvärde] func [namn på funktion]” följt av parametrar, även om funktionen inte har några parametrar ska parenteser skrivas

```
def int func calc(int a, int b)
{
      return a+b;
};

def int func print_hej()
{
      print(”Hej!”)
};

print(calc(1,2));
>> 3
```

```
x = 5
y= 1
if (x<3) {
      print(”a”);
};
else if (x=5 && y==2) {
      print(”b”);
};
else {
      print(”c”);
};
output:
c
```
