

# Seeds of Programming

*"I have the right to demand obedience because my orders are reasonable ones"*

*The Little Prince*

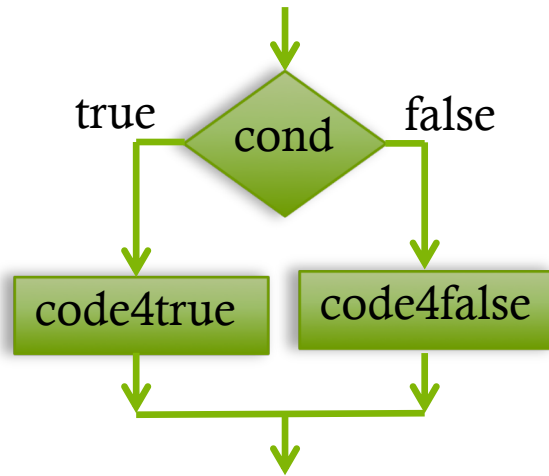


# Write a program

- ◆ A program is a sequence of instructions
- ◆ The flow of the instructions can be changed in response of conditions.
- ◆ Programs manipuates **variables** (the main in Web is the DOM)
- ◆ The main control structures are:
  - ◆ if-then-else
  - ◆ while
  - ◆ for
- ◆ Instructions can be packed in **functions** and **procedures**.
- ◆

# if-then-else

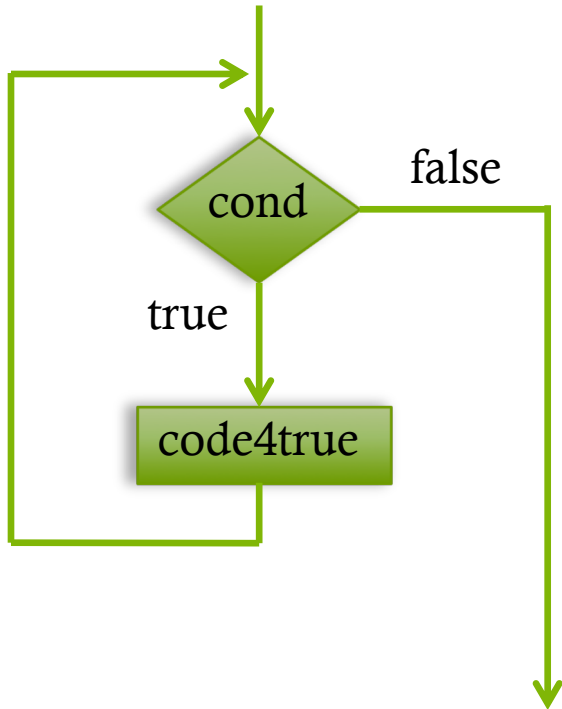
- if a condition is true then exec a specific piece of code, otherwise exec another piece of code



```
...  
if (cond==true) {  
    do_this_because_true();  
}else{  
    do_this_because_false();  
}  
...
```

# While

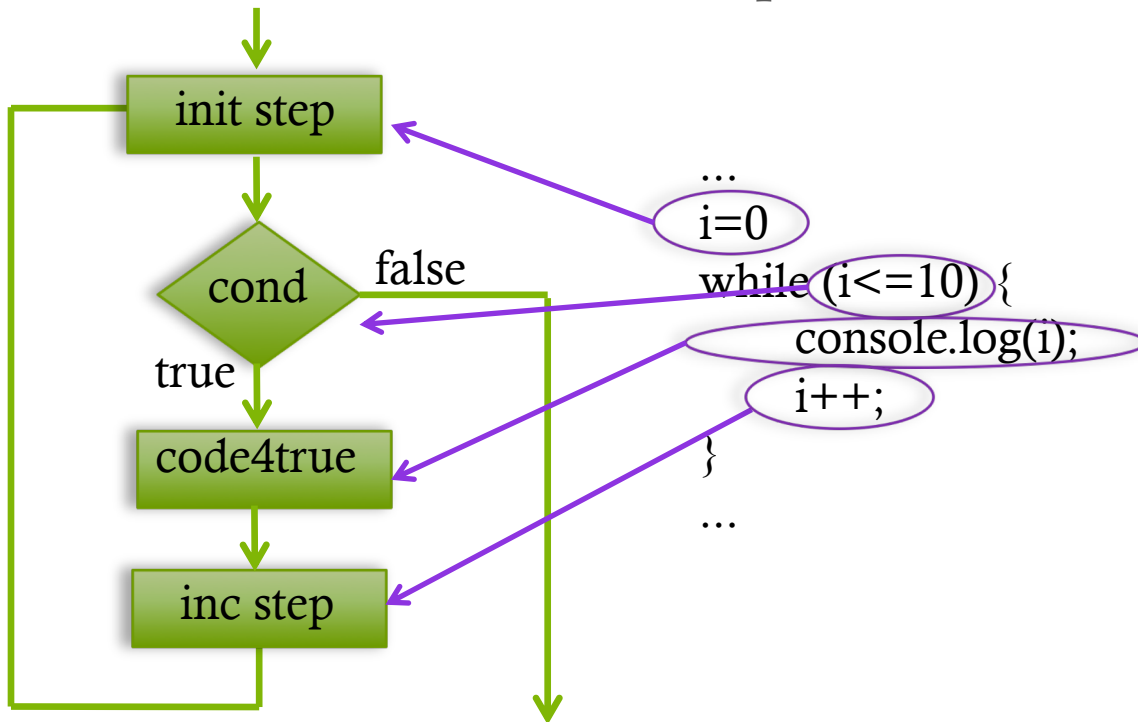
- ◆ Exec (may be never) a specific code if a condition is true



```
...  
while (cond==true) {  
    do_this_because_true();  
}  
...
```

# For

- Initialize something, if a condition is true exec a specific code and do a "incremental" step

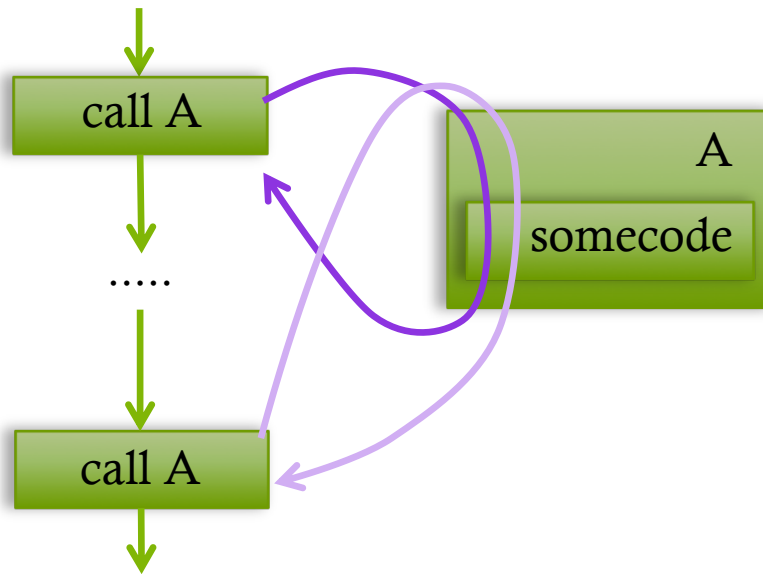


*useful for repeat n times  
a specific code*

```
...  
for (i=0; i<=10;i++) {  
    console.log(i);  
}  
...
```

# Procedures

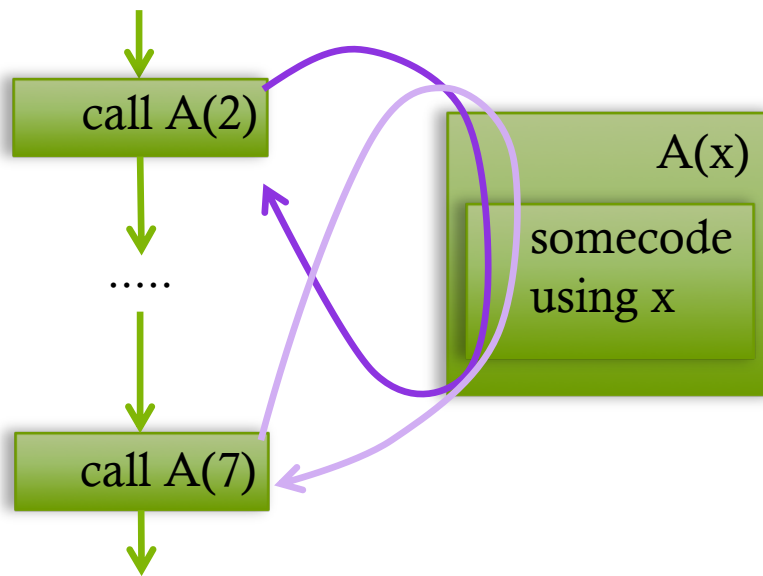
- ◆ A procedure is a pack of instructions that can be called from another point of the program



```
...  
procedure A() {  
    print("execute A");  
}  
...  
call A();  
...  
...  
call A();  
....
```

# Procedures with arguments

- Procedure can have call parameters, i.e. local variables that can be instantiated with a given value during the call



```
...  
procedure A(x) {  
    print("x is equal to ",x);  
}
```

```
...  
call A(2);
```

```
...
```

```
...  
call A(7);
```

```
....
```

*This code outs:*

*"x is equal to 2  
x is equal to 7"*

# Scope

- ◆ The scope of a variable is the portion of the code that can be access to it.

*Usually (depending on programming languages) global variable are accessible to anywhere*

```
...  
var x=1;  
...  
procedure outX(){  
    print("x is equal to ",x);  
}  
...  
outX();  
....
```

*This code outs:  
"x is equal to 1"*



# Scope with procedures

*Usually (depending on programming languages) local variables are only accessible to the procedure who creates it and to the procedures called from it (and only during the call).*

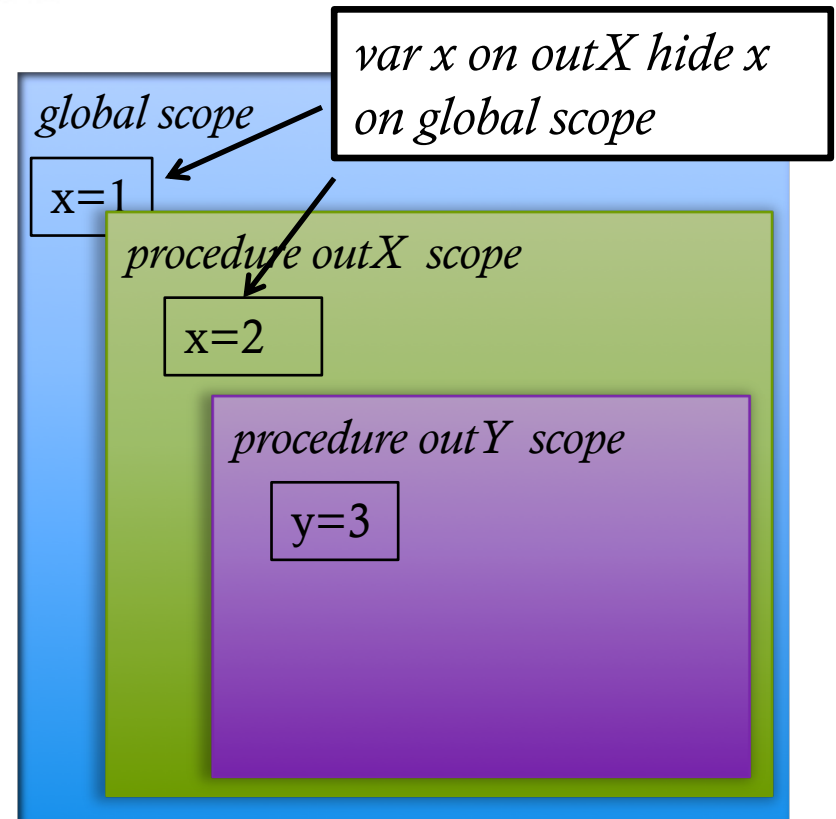
```
...  
var x=1;  
...  
procedure outX(){  
    var x=2;  
    print("x is equal to ",x);  
}  
...  
outX();  
....
```

*This code outputs:  
"x is equal to 2"*

# Scope with sub-procedures

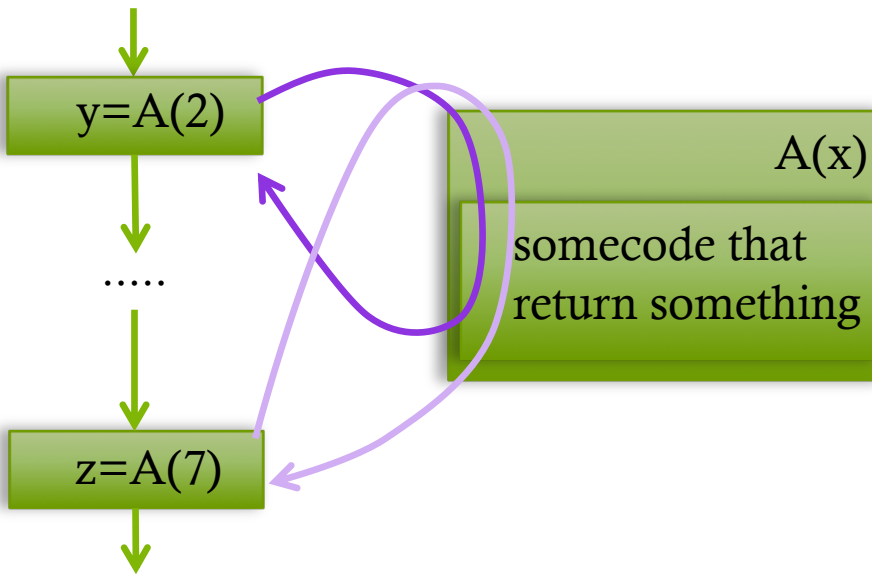
```
...  
var x=1;  
...  
procedure outX(){  
    var x=2;  
    print("x is equal to ",x);  
    outY(x+1)  
}  
...  
procedure outY(y){  
    print("y is equal to ",y);  
}  
...  
outX();  
....
```

*This code outs:  
"x is equal to 2  
y is equal to 3"*



# Functions

- ◆ A function is a procedure that return some values



```
...  
function A(x) {  
    return x+1;  
}  
...  
y=A(2);  
print ("y is equal to ",y);  
...  
z=A(7);  
print ("z is equal to ",z);  
....
```

*This code outs:  
"y is equal to 3  
z is equal to 8"*