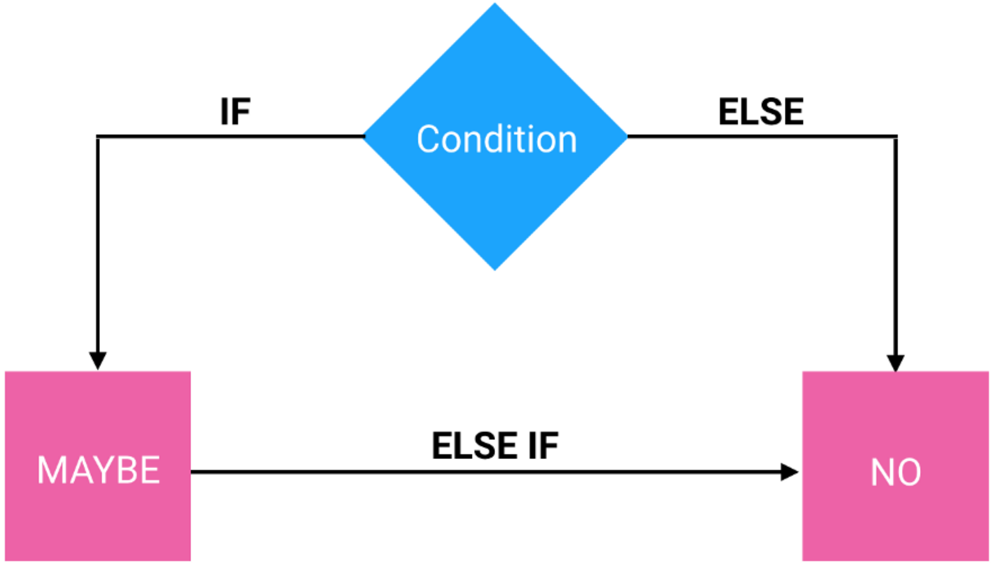
Spatial Simulation

Winter Semester 2023 / 24, MSc Applied Geoinformatics

Principle of coding (with GAMA) II:

Conditions | Variables (advanced) | Loops

Conditions



Conditional statements

Now, our lions shall get mature and die, depending on their age:

```
if(age > 2) {
    write "I am mature";
} else if (age > 60) {
    write "I will die";
    do die;
} else {
    write "Error: I should be dead now!";
}
```

Conditionals

What is the result for x = 4?

```
x <- rnd(5);
if x = 4 { //4
    write "x is exactly 4";
}
else if x >= 2 { //2, 3, 4
    write "x is equal or greater than 2";
}
else {
    write "x is something else";
}
```

Result for 4: exactly 4

```
x <- rnd(5);
if x = 4 { //4
    write "x is exactly 4";
}
if x >= 2 { //2, 3, 4
    write "x is equal or greater than 2";
}
else {
    write "x is something else";
}
```

Result for 4: exactly 4 & equal or greater than 2

Conditional statements

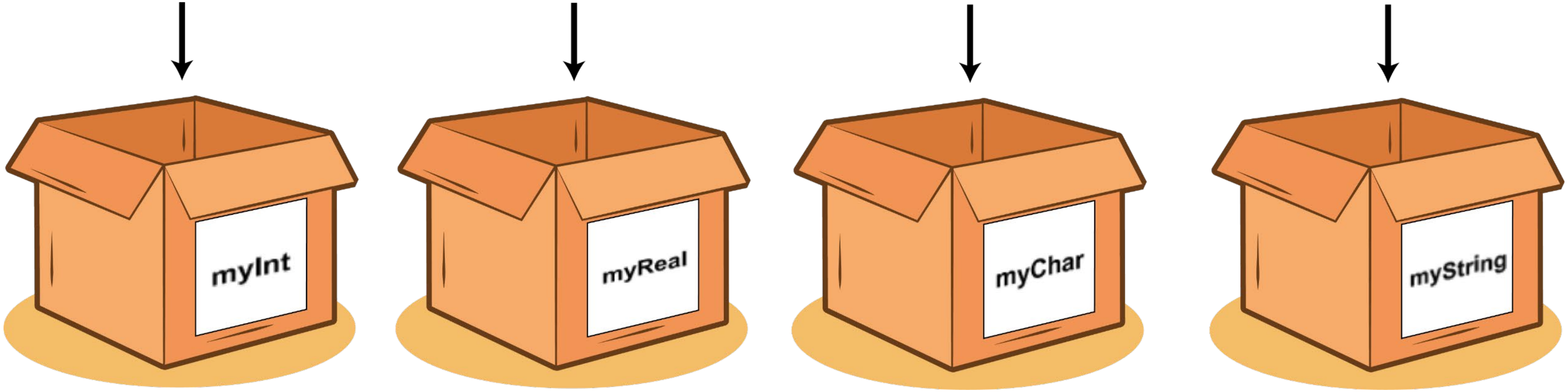
To execute a reflex only, if some condition is true, use when:

For example, this lion keeps falling in love every time step, once it gets mature:

```
reflex fall_in_love when: is_mature = true {
}
```

Variables

4 2.5 “a” “hello”



int float string

Variables types in GAMA

Declaration in GAMA	Type	Example
int	integer	1, 2, 153
float	A real number	0.563, 5.6, 13.584368
string	characters	“hello”, “a”, “this is great”
bool	boolean	True / False
list	List of elements	<a, b, c>, <5, 9 , 3, 10>
rgb	a color in the RGB space	rgb(3,5,67), #red
file	an image or data file	.txt, .csv, .tif, .gif, .asc, .shp
geometry	point, line, or polygon	circle(2), point([2,5])
graph	graph with nodes / edges	[node(0), node(1), node(2)]

Variables – complex types: lists

Declare a global variable of type list that contains variables of type integer and assign an empty list

```
    list<int> age_list <- [];
```

Add values to / remove values from the list

```
    add age to: age_list;
```

Compute the mean value of the list in the global section; see also: length(), min(), max():

```
    write “The mean age of lions is: “ + mean(age_list);
```

Variables – built-in variables

You don‘t have to declare them, they are available for any species

```
   name
   location
```

or when you define a skill.

```
skills:[moving]
   speed
   heading
```

Calculate with complex variables: rgb

Declare a colour variable

```
rgb my_colour;
```

To represent a colour with a variable of type rgb, write:

```
my_colour <- rgb(60,100,10); //max value of rgb values = 255
my_colour <- #cyan;
```

You can also use the colour variable to visualise another variable:

```
my_colour <- rgb(0,age,0);
```

To visualise your agent in that colour:

```
aspect default {
    draw circle(2) color: my_colour;
}
```

Pseudo-Variable „self“

Refers to the current agent.

```
write self;
```

Returns, for example:

```
lions(3)
```

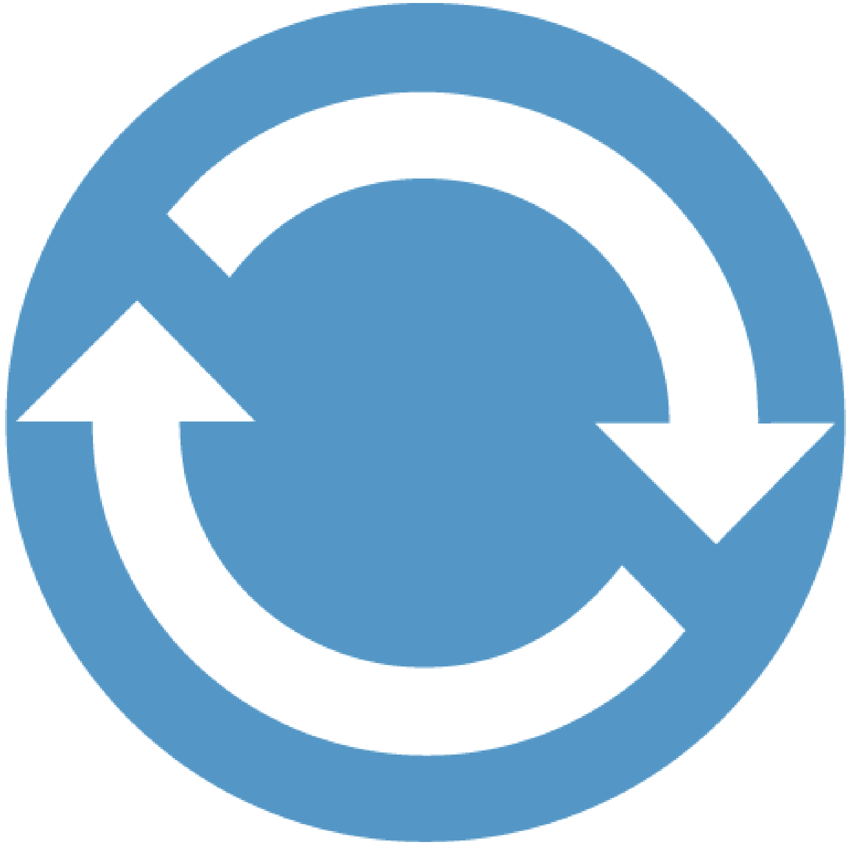
Pseudo-Variable „myself“

Myself refers to the calling agent (one hierarchy level above).

```
species lions {
    reflex reporting {
        ask lions(3) {
            write "This is the current agent: " + self; //lions(3)
            write "This is the calling agent: " + myself; //lions(0)
        }
    }
}
```

Returns -> ???

Loops



Defining loops

Loop

```
    loop times: 5 {
        write "Hi there!";
    }
```

Reflex within a species loops through all agents

```
    reflex report_age {
        write age;
    }
```

Ask (called e.g. from the global section) loops through all agents:

```
    ask lions {
        write age;
    }
```

Loops

Loop

```
loop times: 5 {
    x <- x + 1;
    write "Hi there " + x + "!";
}
```

Results in

```
Hi there 1!
Hi there 2!
Hi there 3!
Hi there 4!
Hi there 5!
```

Loop through all agents / cells

from the global section: ask agents or cells to do something; loops through all agents:

```
ask lions {
    write age;
}
```

Results in:

```
    5
    8
    9
    4
    9
```

Ask a specific agent

To call a specific lion, use the index number:

```
ask lions(3) {
    write age;
}
```

`Results in:`

```
4
```