دانشکده روانشناسی و علوم تربیتی

پژوهشگاه دانش‌های بنیادی IPM

MATLAB for Brain and Cognitive Psychology (Control Statements)

Presented by:

Ehsan Rezayat, Ph.D.

Faculty of Psychology and Education, University of Tehran,

Institute for Research in Fundamental Sciences (IPM), School of Cognitive Sciences,

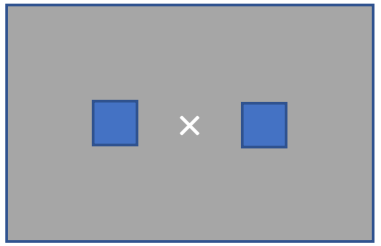
emails: rezayat@ut.ac.ir, rezayat@ipm.ir, erezayat.er@gmail.com

Flow Control

```
Show Fixation  for 1 sec
SOA = -10;  Random selection  rand (1)

Check if  SOA < 0
Show  Left Target
Wait (SOA msec)
Now show  Right Target

Check if  SOA > 0
Show  Right Target
Wait (SOA msec)
Now show  Left Target

Clear Target
Clear Fixation

Subject make selection

Check if the subject Choose Correct choice
```

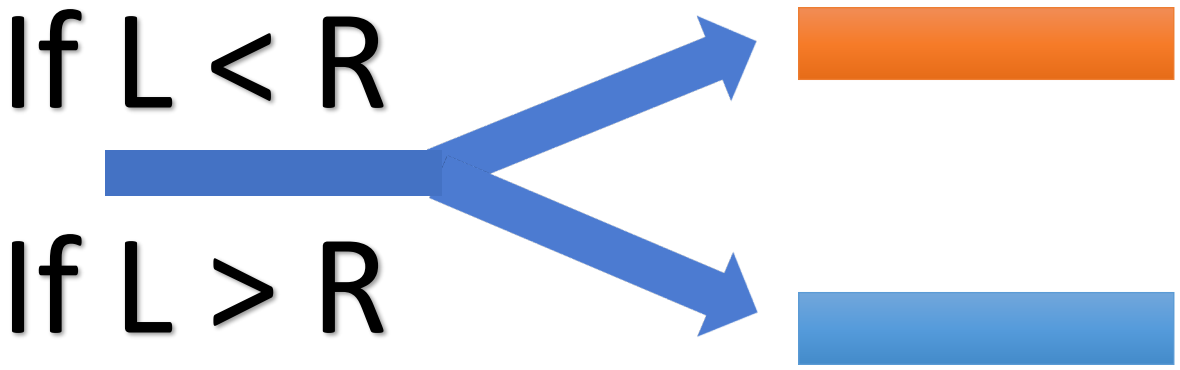
Temporal order Judgment

SOA
-50 :2: 50 ms



Saccades
Left or Right

Conditionals statements




```
Check if  SOA < 0
Show  Left Target
Wait (SOA msec)
Now show  Right Target

Check if  SOA > 0
Show  Right Target
Wait (SOA msec)
Now show  Left Target


Check if the subject Choose Correct choice
```

Conditionals statements

```
if condition
        do this stuff
end
```

condition should evaluate to logical true or false. examples:

x > 5

y == 5

strcmp(subject,'S01')

Conditionals statements

```
if condition
        do this stuff
end
```

```
if condition
        do this stuff
else
        do this stuff
end
```

```
if condition
        do this stuff
elseif condition
        do this stuff
else
        do this stuff
end
```

Conditionals statements

```
switch  variable
        do this stuff
case num1
        do this stuff
case num2
        do this stuff
case num3
        do this stuff
end
```

```
try
        do this stuff
catch
        do this stuff
end
```

Getting the truth

- == equal to (distinguish from = which sets a value)
- ~= not equal to
- > greater than
- < less than
- >= greater than or equal to
- <= less than or equal to

Testing the truth

- Logical operators:
  - && AND (sometimes you will see &)
  - || OR (sometimes you will see |)
  - ~ NOT

AND		
A ∧ *B*		Output
F	F	**F**
T	F	**F**
F	T	**F**
T	T	**T**

OR		
A ∨ *B*		Output
F	F	**F**
T	F	**T**
F	T	**T**
T	T	**T**

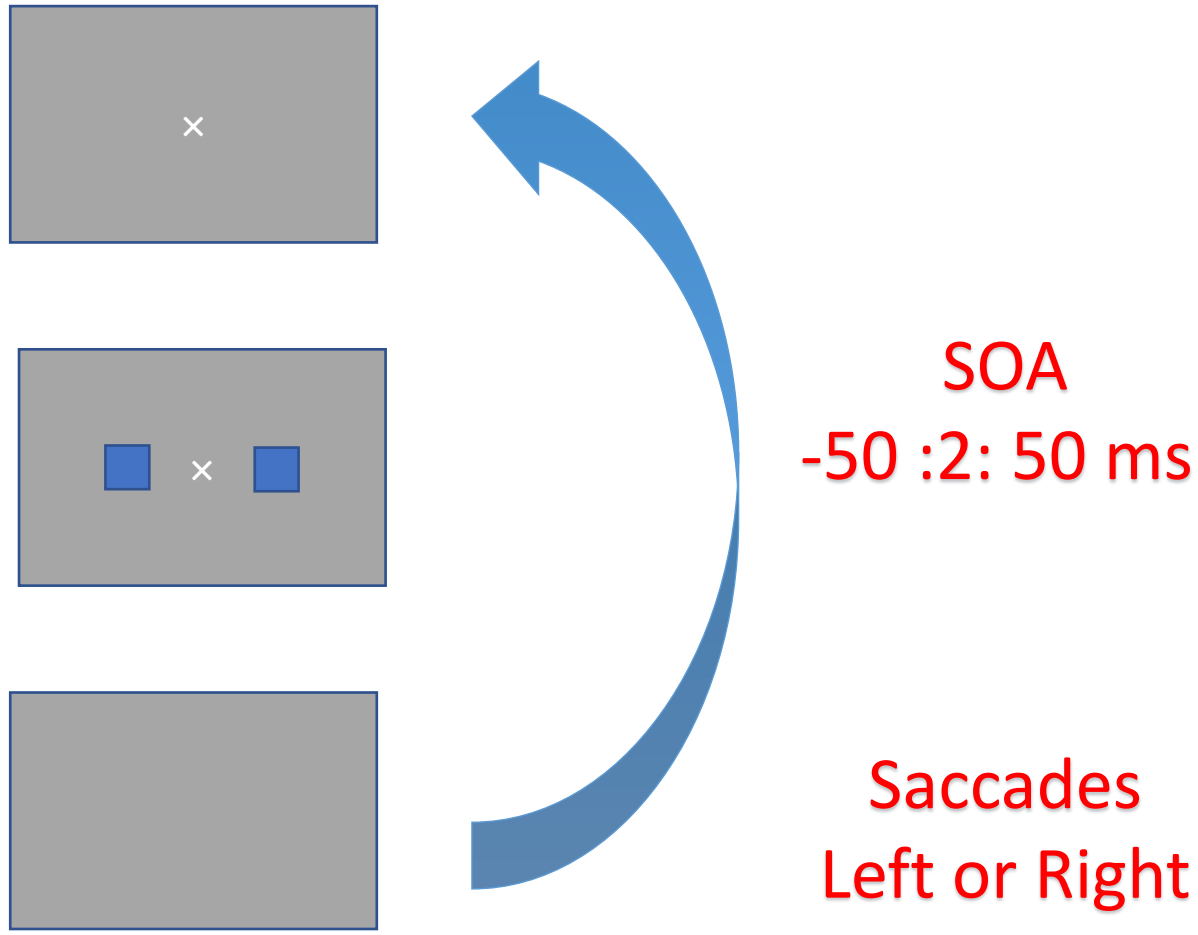
NOT	
¬A	Output
F	**T**
T	**F**

Comparing strings

```
>> x = 'hello';
>> y = 'goodbye';
>> x == y
Error using  ==
Matrix dimensions must agree.

>> help strcmp
strcmp Compare strings.
    TF = strcmp(S1,S2) compares the strings S1 and S2 and returns logical 1
    (true) if they are identical, and returns logical 0 (false) otherwise.

>> strcmp(x,y)
ans =

     0

>> y = 'Hello';
>> strcmp(x,y)

ans =

     0

>> strcmpi(x,y)

ans =

     1
```

Code repetition (200 Trials)

```
Show Fixation  for 1 sec
SOA = -10;  Random selection  rand (1)

Check if  SOA < 0
Show  Left Target
Wait (SOA msec)
Now show  Right Target

Check if  SOA > 0
Show  Right Target
Wait (SOA msec)
Now show  Left Target

Clear Target
Clear Fixation

Subject make selection

Check if the subject Choose Correct choice
```



For loops

```
function doLoop()
%do a loop

for i = 1:10
        j = sqrt(i);
        fprintf('The square root of %d is: %.2f\n',i,j);
end
```

counter variable

range of values for counter to take on

code to be repeated

```
>> doLoop()
The square root of 1 is 1.00
The square root of 2 is 1.41
The square root of 3 is 1.73
The square root of 4 is 2.00
The square root of 5 is 2.24
The square root of 6 is 2.45
The square root of 7 is 2.65
The square root of 8 is 2.83
The square root of 9 is 3.00
The square root of 10 is 3.16
```

For loops

```
function doLoop()
%do a loop

listOfPeople = {'Fred','Mary','Laura'};

for i = 1:length(listOfPeople)
        name = listOfPeople{i};
        fprintf('Person number %d is %s\n',i,name);
end
```

```
>> doLoop()
Person number 1 is Fred
Person number 2 is Mary
Person number 3 is Laura
```

While loops

```
while condition
        do this
stuff
end
```

While loops

```
function doLoop()
%do a loop

x = 0;
while x < 10
        y = x^2;
        fprintf('%d squared is %d\n',x,y);
        x = x + 1;
end
```

```
>> doLoop()
0 squared is 0
1 squared is 1
2 squared is 4
3 squared is 9
4 squared is 16
5 squared is 25
6 squared is 36
7 squared is 49
8 squared is 64
9 squared is 81
```

While loops

Infinite loops

```
function doLoop()
%do a loop

x = 0;

while 1
        x = x + 1;
        fprintf('x is %d\n',x);
end
```

Functions

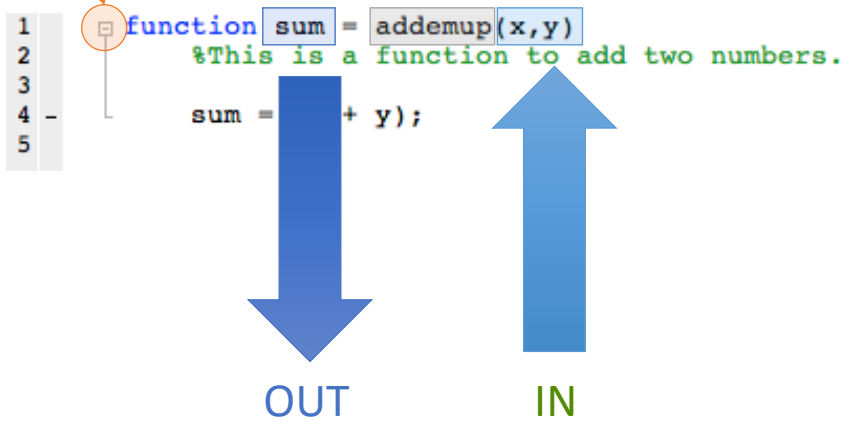
What is a function?

- A function is a self-contained piece of code that accomplishes a specific function
- It may take in certain variables (parameters) and return results

Function declarations

All functions must be *declared*, that is, introduced in the proper way.

code folding



result of the function

name of the function

parameters passed to the function

Function declarations

Functions may return no variables:

```
function printAName(name)
        %Not very exciting.  Just prints a name.

        fprintf('The name is: %s\n',name);
```

Or several:

```
function [avg,biggest,smallest] = getSomeStats(x)
        %Return some statistics on vector x

        avg = mean(x);
        biggest= max(x);
        smallest = min(x);
```

Variable scope

- Variables only exist within a certain “scope”
- Variables defined in a function only exist within that function

Variable scope

```
1   function sum = addemup(x,y)
2       %This is a function to add two numbers.
3
4 -     multiplier = 3;
5 -     sum = (x + y) * multiplier;
6
7 -   return;
8
```

```
1   function sum = addemup2(x,y)
2       %This is a function to add two numbers.
3
4 -     multiplier = 5;
5 -     sum = (x + y) * multiplier;
6
7 -   return;
```

```
>> addemup(1,1)
ans =
     6
>> addemup2(1,1)
ans =
    10
```

Debugging

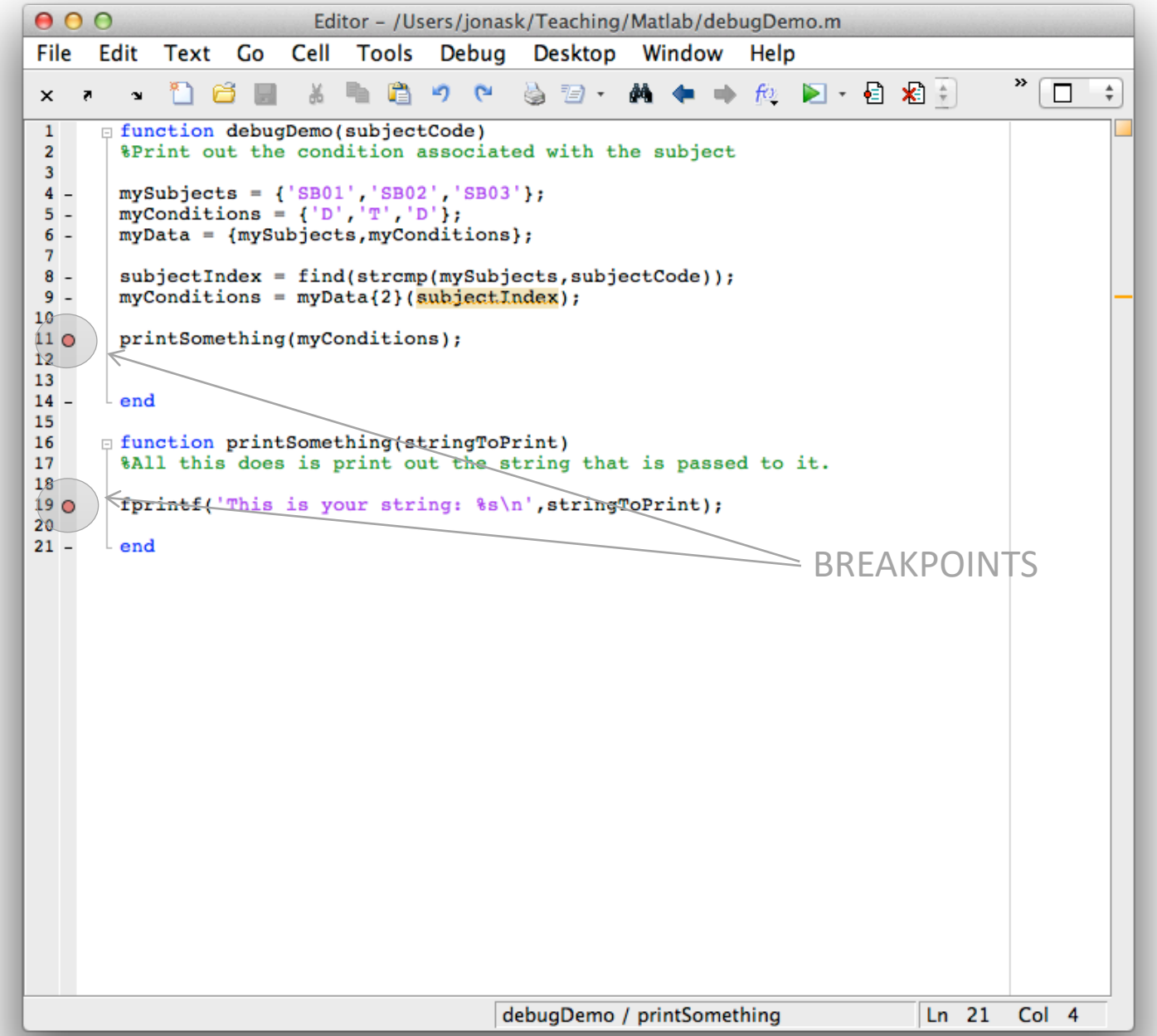
```
>> debugDemo('SB02')
Error using fprintf
Function is not defined for 'cell' inputs.

Error in debugDemo>printSomething (line 19)
fprintf('This is your string: %s\n',stringToPrint);

Error in debugDemo (line 11)
printSomething(myConditions);
```

proximal cause of error

distal cause of error

Links to the help file for that function

Links to the line in the script where the problem occurred


Editor – /Users/jonask/Teaching/Matlab/debugDemo.m
File Edit Text Go Cell Tools Debug Desktop Window Help
function debugDemo(subjectCode)
%Print out the condition associated with the subject
mySubjects = {'SB01','SB02','SB03'};
myConditions = {'D','T','D'};
myData = {mySubjects,myConditions};
subjectIndex = find(strcmp(mySubjects,subjectCode));
myConditions = myData{2}(subjectIndex);
printSomething(myConditions);
end
function printSomething(stringToPrint)
%All this does is print out the string that is passed to it.
fprintf('This is your string: %s\n',stringToPrint);
end
BREAKPOINTS
debugDemo / printSomething
Ln 21 Col 4

Debugging

```
>> debugDemo('SB02')
11   printSomething(myConditions);
K>>
```

K>> prompt indicates that are are in debug mode

This is the line where the script has paused (we set a breakpoint here). The line number is a link to the line in the editor.

Workspace shifts to showing all the variables in memory inside the function

Assignment session #3

- Write a function named “yourInitials_session3()”
- The function should take two inputs:
  1) a string containing the subject’s code
  2) a vector of 5 scores
- The function should return two values:
  1) the mean of the 5 scores, after removing the lowest one
  2) the standard error of the mean of the 5 scores after removing the lowest one
- The function should also do the following when run:
  print the following line to the screen:
  “Working on subject XXXX…” where XXXX is the subject code