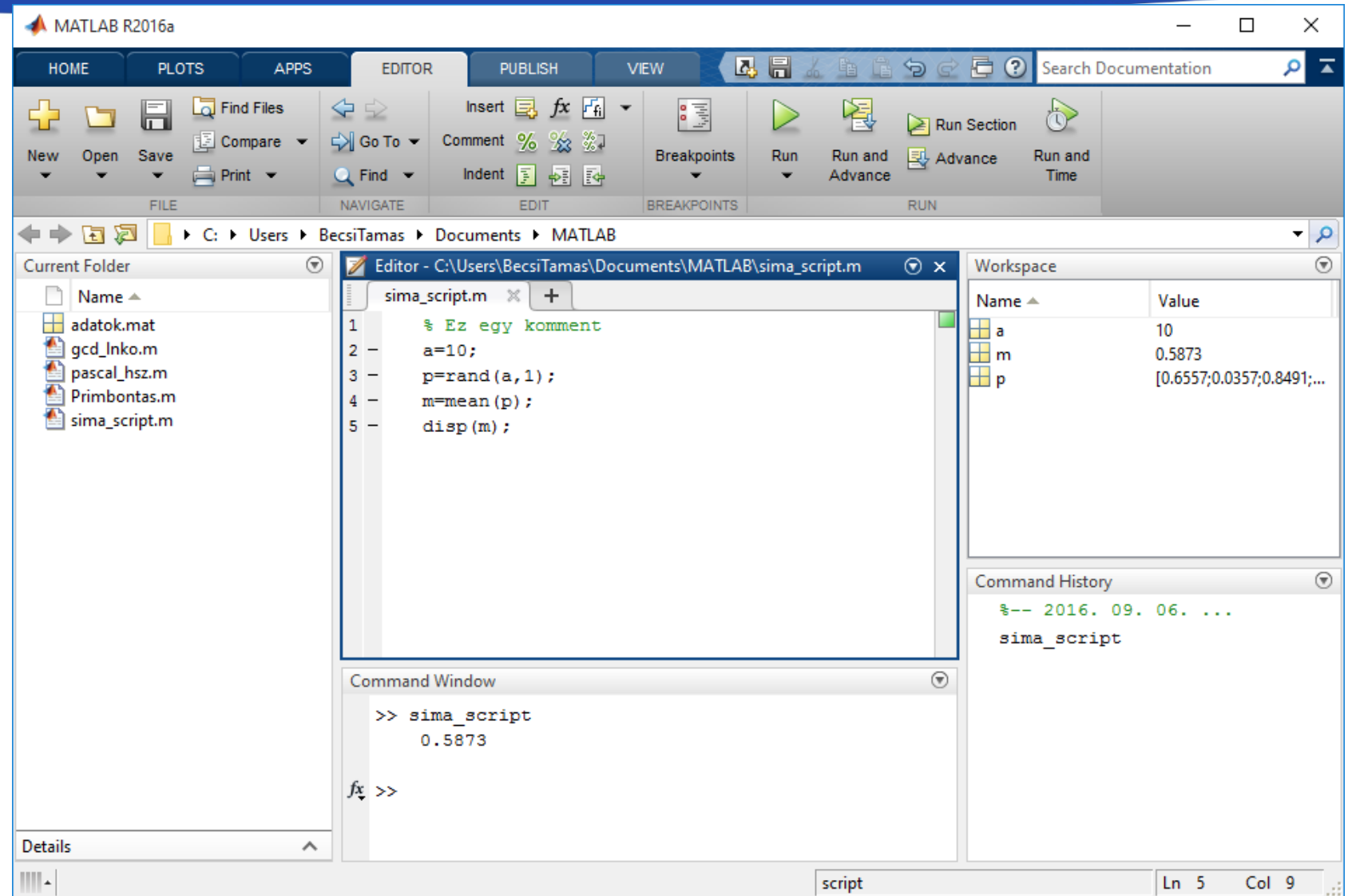


MATLAB OKTATÁS
2. ELŐADÁS
FLOW CONTROL

Dr. Bécsi Tamás
Hegedüs Ferenc

SCRIPTEK, CONTROL FLOW

New
->Script



The screenshot displays the MATLAB R2016a environment. The main window is titled "MATLAB R2016a" and shows the "EDITOR" tab. The current folder is "C:\Users\BecsiTamas\Documents\MATLAB". The script "sima_script.m" is open in the editor, containing the following code:

```
1 % Ez egy komment
2 a=10;
3 p=rand(a,1);
4 m=mean(p);
5 disp(m);
```

The workspace shows the following variables:

Name	Value
a	10
m	0.5873
p	[0.6557;0.0357;0.8491;...

The Command Window shows the execution of the script:

```
>> sima_script
0.5873

fx >>
```

The Command History shows the command "sima_script" was executed on 2016.09.06.

Syntax

if *expression*

statements

[elseif *expression*

statements]

[else

statements]

end

```
a=10;
p=rand(a,1);
m=mean(p);
disp(m);
if m<0.45
    disp('Az átlagérték kisebb a vártnál');
elseif m>0.55
    disp('Az átlagérték nagyobb a vártnál');
else
    disp('Az átlagérték megfelelő');
end;
```

SWITCH, CASE, OTHERWISE

Syntax

```
switch switch_expression
  case case_expression
    statements
  case case_expression
    statements
  ...
  otherwise
    statements
end
```

```
n = input('Enter a number: ');

switch n
  case -1
    disp('negative one')
  case 0
    disp('zero')
  case 1
    disp('positive one')
  otherwise
    disp('other value')
end
```

FOR

Syntax

```
for index = values  
    statements  
end
```

```
sum=0;  
for i=1:10  
    sum=sum+i;  
end;  
disp(sum);
```

```
%Szorzótábla  
a=zeros(10);  
for i=1:10  
    for j=1:10  
        a(i,j)=i*j;  
    end  
end;
```

FOR 2

```
%Sorozat
for v = 1.0:-0.2:0.0
    disp(v)
end
```

```
1
0.8000
0.6000
0.4000
0.2000
0
```

```
%Vektor
for v = [1 5 8 17]
    disp(v)
end
```

```
1
5
8
17
```

```
%Mátrix:
for I = eye(4,3)
    disp('Current unit vector:')
    disp(I)
end
```

Current unit vector:

```
1
0
0
0
```

Current unit vector:

```
0
1
0
0
```

...

WHILE

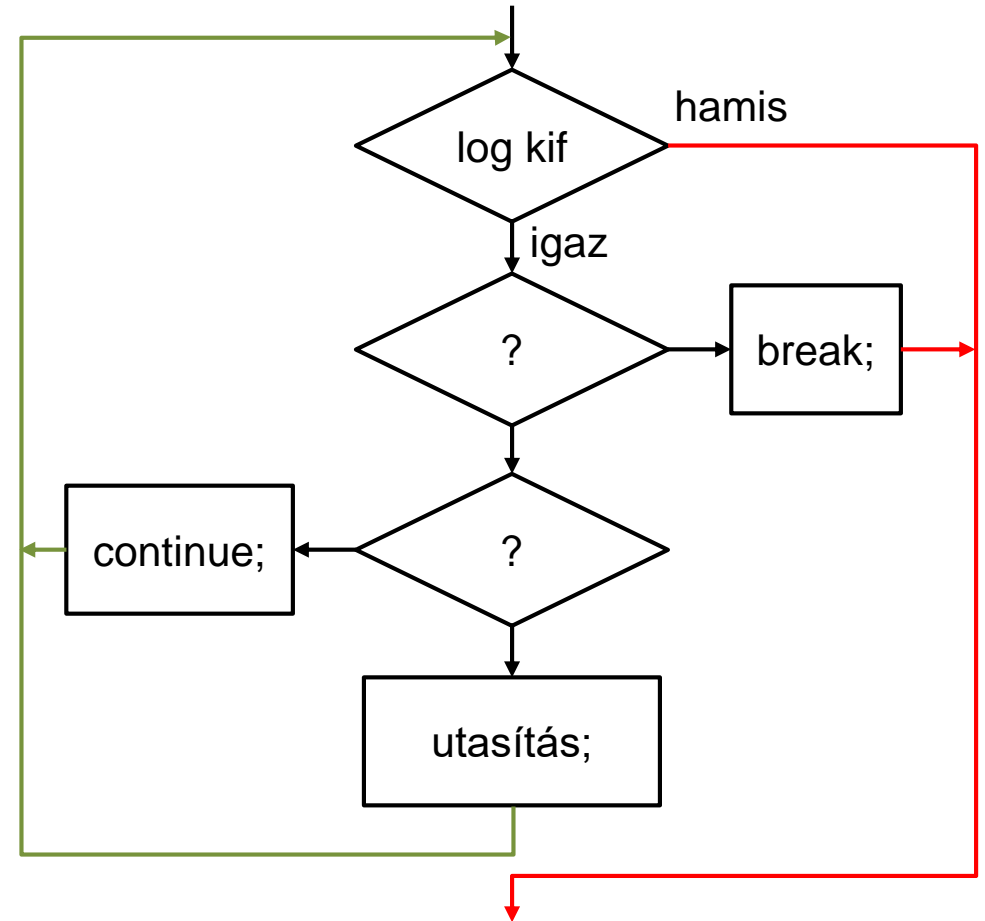
Syntax

```
while expression  
    statements  
end
```

```
n = 10;  
f = n;  
while n > 1  
    n = n-1;  
    f = f*n;  
end  
disp(['n! = ' num2str(f)])
```

BREAK, CONTINUE

- A ciklusok működését befolyásolhatjuk a `break` és a `continue` utasításokkal
- A *break* azonnal kilép a ciklusból
- A *continue* visszaugrik a ciklus elejére (a bennmaradási feltétel elé)



BREAK, CONTINUE PÉLDÁK

```
limit = 0.8;
s = 0;

while 1
    tmp = rand;
    if tmp > limit
        break
    end
    s = s + tmp;
end
```

```
for n = 1:50
    if mod(n,7)
        continue
    end
    disp(['Divisible by 7: '
num2str(n)])
end
```

PAUSE, RETURN

- A return utasítás megszakítja a futó script, vagy függvény futását, és visszaadja a vezérlést az azt meghívó egységnek.
- A pause utasítás gombnyomásig (vagy az argumentumban átadott másodpercig) felfüggeszti a script futását

```
for i=1:10
    disp(i);
    pause(0.5);
    if mod(i,7)==0
        pause;
    end
end
```

TRY CATCH

Syntax

```
try
    statements
catch exception
    statements
end
```

```
a=ones(3);
b=ones(4);
try
    c=a*b
catch ME
    disp('Ezt így, ebben a
formában, nem lehet');
end
```

Syntax

```
function [y1,...,yN] = myfun(x1,...,xM)
```

- Függvények szerepelhetnek külön file-ban, (ekkor a file neve megegyezik a függvény nevével)
- Az argumentumok, és a visszatérési érték is lehet egy változó, vagy vektor

FUNCTION PÉLDÁK

```
function y = average(x)
y = sum(x)/length(x);
end
```

```
>> average(1:99)
```

```
ans =
```

```
50
```

```
function [m,s] = stat(x)
n = length(x);
m = sum(x)/n;
s = sqrt(sum((x-m).^2/n));
end
```

```
>> [m,s]=stat(1:99)
```

```
m =
```

```
50
```

```
s =
```

```
28.5774
```