

User-Defined Functions

```
close all  
clear all  
- - -  
max(x);  
mean(1);  
sqrt(1);
```

step 1: Create a new script file (No 'close all', No 'clear all' there), deg2rad.m file name

```
function r = deg2rad(deg)  
r = deg * pi / 180;  
end
```

step 2: start the main script (a new script)
in the same directory.

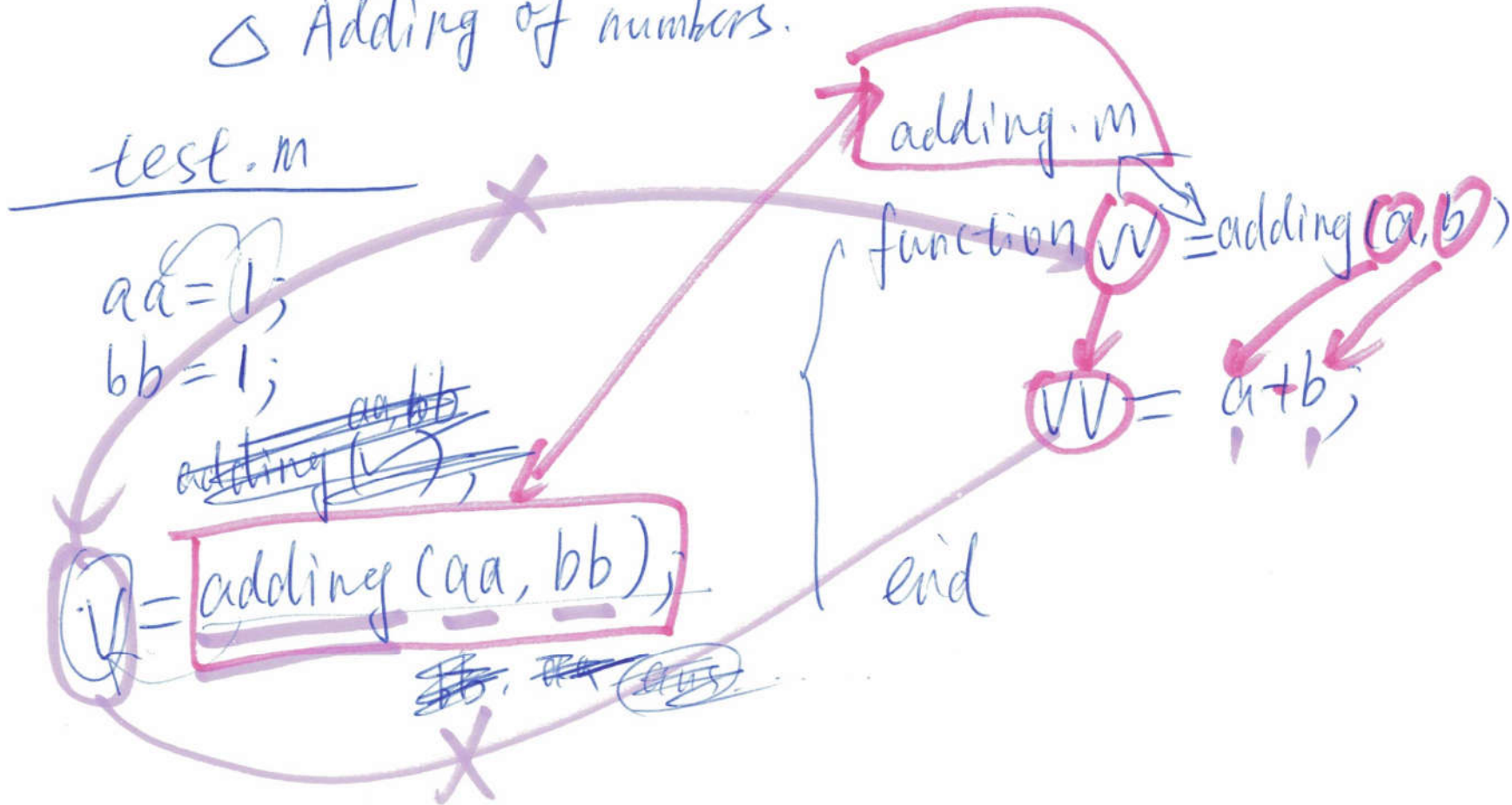
test.m

close all
clear all

ans = deg2rad(30);

r

△ Adding of numbers.



△ put the test.m file and the adding.m file in the same directory.

③

Naming a function file.

- △ letters, digits, underscores.
- △ Follows the same rules as variable names
- △ No spaces, No dashes.
- * avoid using names of built-in functions

Try: calculate the circle area

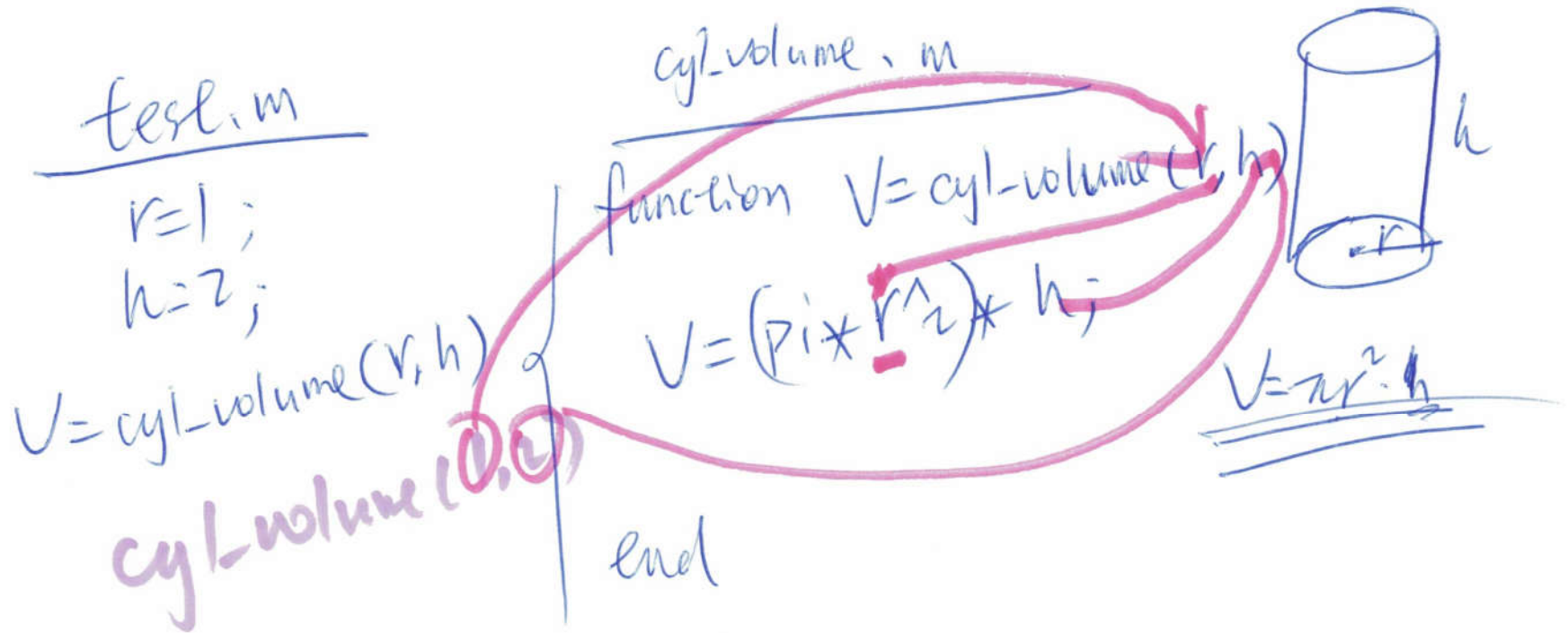
test.m

```
r = 200;  
a = area(r);
```

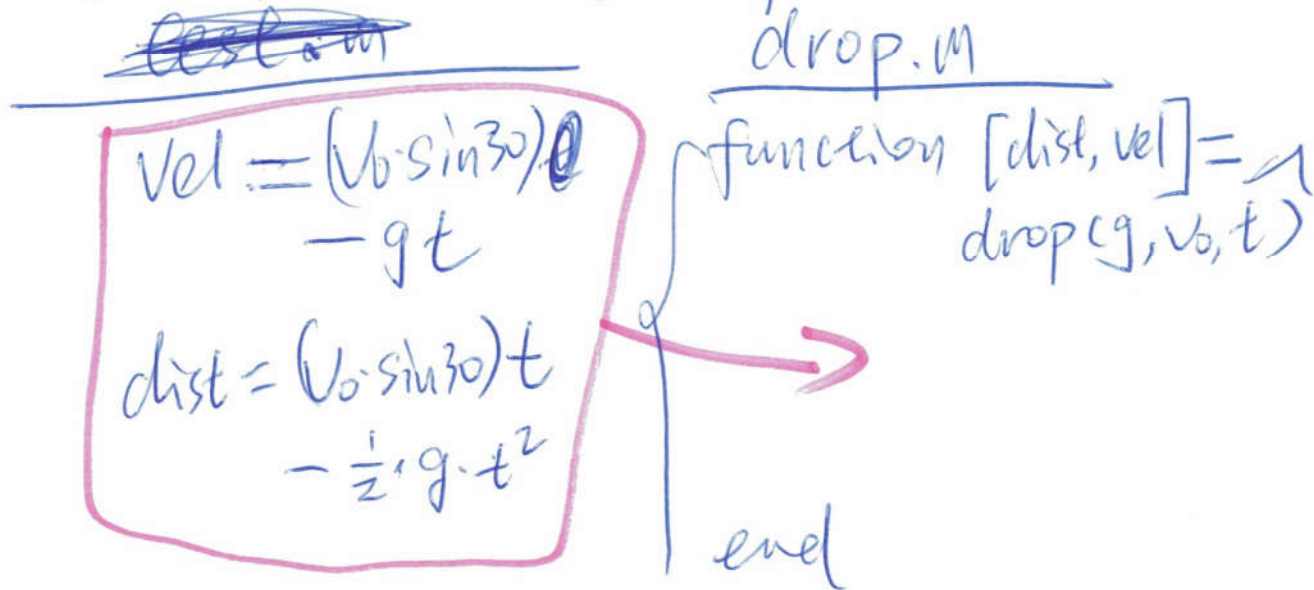
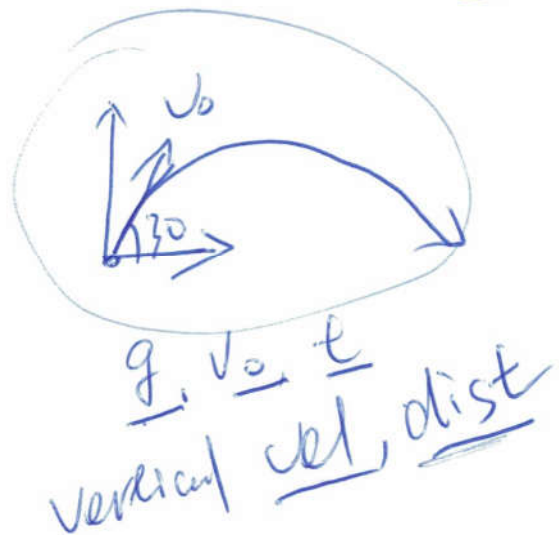
area.m

```
function a = area(r)  
    a = pi * r^2;  
end
```

△ Two inputs, one output: volume of a cylinder



△ multiple inputs, multiple outputs.



5

test.m

$$g = 9.8; \quad v_0 = 30; \quad t = 10;$$

$$[dist, vel] = \underline{drop}(g, v_0, t);$$

$$[max, ind] = \max(x)$$

tasks: use-defined functions for max(x)

test.m

$$X = [\dots];$$

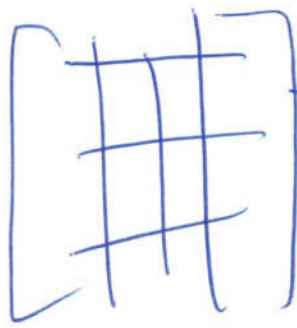
$$mm = \maxXX(X);$$

maxXX.m

```
function mm = maxXX(x)
    mm = x(1);
    for i = 1: length(x)
        if x(i) > mm
            mm = x(i);
        end
    end
end
```

(b)

4. Find the max in a 2-D array.



test.m

X = [] ;

Spring-Break = maxx(x);

maxxx @.m

mm = x(1,1);

```
function mm = maxx(x);  
for i = 1:length(x(:,1))  
    for j = 1:length(x(i,:))  
        if x(i,j) > mm  
            mm = x(i,j);  
        end  
    end  
end  
end
```

test.m

```
E = 1; (n = 1);  
while (E > 1e-8)  
    Result = appro_pi((M));  
    (E) = abs(Result - pi);  
    m = m + 1;  
    2  
end
```

appro_pi.m

```
function Result Result = appro_pi(m)  
    S = 0;  
    for i = 1:m  
        S = S + ((-1/3)^(i-1)) / (2*i + 1);  
    end  
    end → Result = sqrt(12) * S;
```