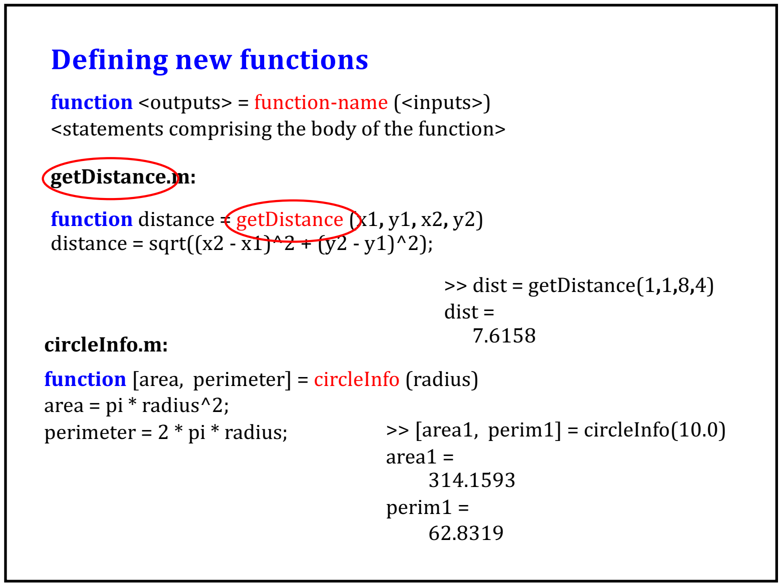


Defining new functions

```
function <outputs> = function-name (<inputs>)
<statements comprising the body of the function>
```

getDistance.m:

```
function distance = getDistance (x1, y1, x2, y2)
distance = sqrt((x2 - x1)^2 + (y2 - y1)^2);
```

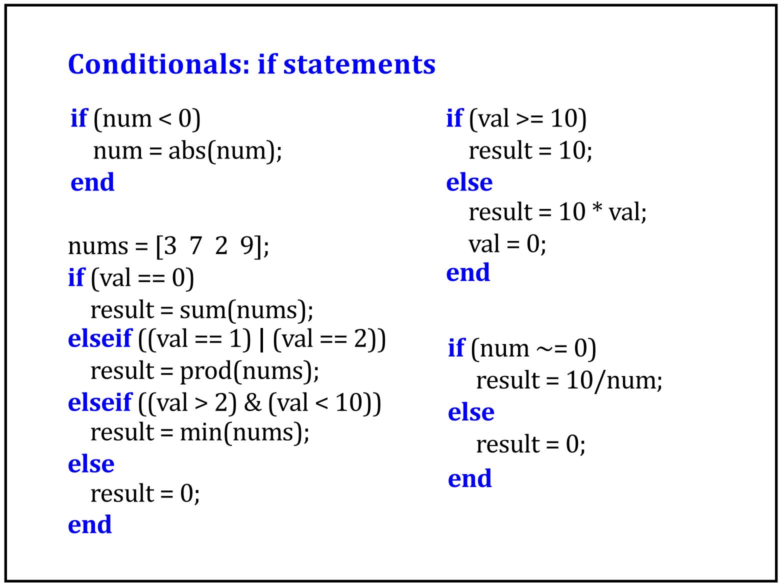
```
>> dist = getDistance(1,1,8,4)
dist =
    7.6158
```

circleInfo.m:

```
function [area, perimeter] = circleInfo (radius)
area = pi * radius^2;
perimeter = 2 * pi * radius;
```

```
>> [area1, perim1] = circleInfo(10.0)
area1 =
      314.1593
perim1 =
      62.8319
```

1



Conditionals: if statements

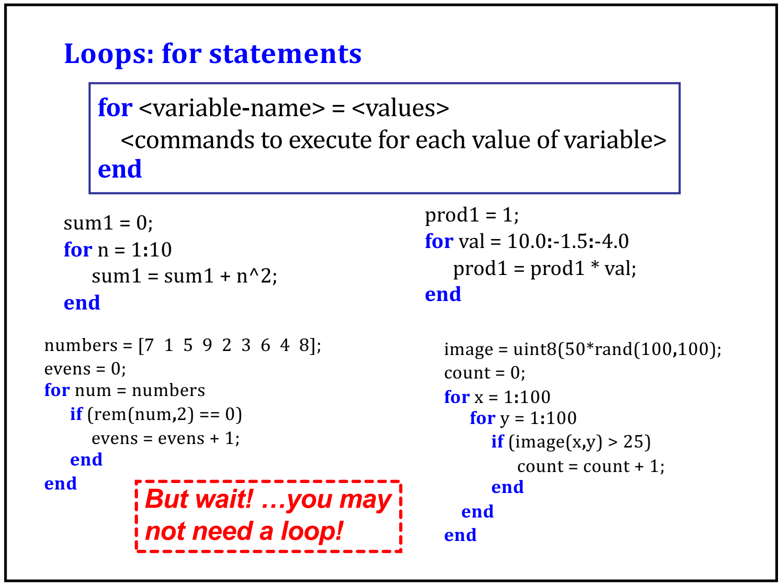
```
if (num < 0)
   num = abs(num);
end

nums = [3  7  2  9];
if (val == 0)
   result = sum(nums);
elseif ((val == 1) | (val == 2))
   result = prod(nums);
elseif ((val > 2) & (val < 10))
   result = min(nums);
else
   result = 0;
end
```

```
if (val >= 10)
   result = 10;
else
   result = 10 * val;
   val = 0;
end

if (num ~= 0)
   result = 10/num;
else
   result = 0;
end
```

2



Loops: for statements

```
for <variable-name> = <values>
   <commands to execute for each value of variable>
end
```

```
sum1 = 0;
for n = 1:10
    sum1 = sum1 + n^2;
end
```

```
prod1 = 1;
for val = 10.0:-1.5:-4.0
    prod1 = prod1 * val;
end
```

```
numbers = [7  1  5  9  2  3  6  4  8];
evens = 0;
for num = numbers
    if (rem(num,2) == 0)
        evens = evens + 1;
    end
end
```

```
image = uint8(50*rand(100,100));
count = 0;
for x = 1:100
    for y = 1:100
        if (image(x,y) > 25)
            count = count + 1;
        end
    end
end
```

But wait! …you may not need a loop!

3