Before we begin

**QZ1**  Online quiz on CMS

**HW1**  Due June 5, 6pm

**OH**  Office hours time & location
Today

- Branching using if/else/end
- Looping using while
- Initialization
- Rounding numbers
- Random numbers
- Formatted print
Branching

- if/else/end
if/else/end

Based on conditions different code fragments are executed

Usage

```plaintext
if %<condition>
  %<statements executed when
  % the condition is TRUE>
else
  %<statements executed when
  % the condition is FALSE>
end
```
number = fix(10*rand);

guess = input('enter a digit: ');

if number == guess
    disp('that is my number!');
else
    if number > guess
        disp('my number is greater');
    else
        disp('my number is smaller');
    end
end
More Conditions

Usage

```plaintext
if %<condition1>
  %<statements executed when
  % condition1 is TRUE>
else
  if %<condition2>
    %<statements executed when
    % condition1 is FALSE and
    % condition2 is TRUE>
  else
    %<statements executed when
    % condition1 is FALSE and
    % condition2 is FALSE>
  end
end
```
Usage

```plaintext
if %<condition1>
    %<statements executed when % the condition1 is TRUE>
elseif %<condition2>
    %<statements executed when % the condition1 is FALSE % and condition2 is TRUE>
else
    %<statements executed when % the condition1 is FALSE % and condition2 is FALSE>
end
```
Loops

- while
While Loop

while
the condition is true, keep executing the same code block

Usage

while %<condition>
%<this code block will be executed
% while the condition is true, until
% it turns into false>
end
What is the sum of numbers from 1 to $n$?

$sum = 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$

Let’s sum these numbers using while-loop.
Compute the sum: \( 1 + \ldots + n \)

sum_numbers.m

```matlab
n = input('enter n: '); % Get the maximum number

j = 1; s = 0; % Initialize counter and sum variables

if n >= 1
    while j <= n
        s = s + j;
        j = j + 1;
    end
else
    error('n should be positive');
end

fprintf('The sum is %d \n', s);
```
numbergame.m

number = fix(10*rand);
guess = -1;

while guess ~= number
    guess = input('enter a digit: ');
    if number == guess
        disp('that is my number!');
    else
        if number > guess
            disp('my number is greater');
        else
            disp('my number is smaller');
        end
    end
end
Prime number
A positive integer that has no positive divisor other than 1 and itself. If $x$ is prime and $x = a \times b$, where $a, b$ are positive integers, $a$ and $b$ have to be either 1 or $x$.

Fact
If $x$ is not a prime, it should have a divisor less than or equal to $\sqrt{x}$ and greater than 1. (Why?)

isprime
Let’s write a function to check if a number is prime or not.
Function: `isprime(n)`

- Returns `true` if `n` is prime, otherwise returns `false`.
- Assumes `n` is a positive integer.

```matlab
function p = isprime(n)
    % Returns true if n is prime, otherwise returns false.
    % Assumes n is a positive integer.

    if n == 1, p = false; return; end

    m = floor(sqrt(n));

    j = 2; p = true;

    while j <= m
        if rem(n,j) == 0, p = false; end
        j = j + 1;
    end
end
```