- A .* B

  mul every ele in A with corresponding every ele in B.

- A ./ B

- ones(3, 1) = $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

- V = $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$; V+1 = $\begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$

- inverse matrix (x^{-1}) :

  pin V

- for loop:

```
for i = 1:10
    v(i) = 2^i;
end;
```

(loop from 1 to 10)

- While loop;

①
```
i=1
while i <= 5
    v(i) = 100;
    i = i+1
end;
```

② i=1 (while + if + break).

```
While true.
    v(i) = 999;
    i = i+1;
    if i == 6,
        break;
    end;
end;
```

- if , else. , elseif.

```
if v(1) == 1
    disp('Yes');
elseif ...
    ---;
else
    ---
end;
```

- fuction.

```
function y = squareThisNumber(x)
y = x^2;
end;
>> squareThisNumber(5)
```

y: or return.

squareThisNumber: function name

x: input.

- pwd

show current path.

```
- function [y1, y2] = squareNumandCube(x)
    y1 = x^2;
    y2 = x^3;
end;
```

Vectorization

Vectorization example.

$$h_\theta(x) = \sum_{j=0}^{n} \theta_j x_j$$

$$= \theta^T x$$

$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$ $x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$

Unvectorized implementation

```
prediction = 0.0;
for j = 1:n+1,
prediction = prediction +
             theta(j) * x(y)
end;
```

Vectorized implementation

```
prediction = theta' * x;
```

Without vectors