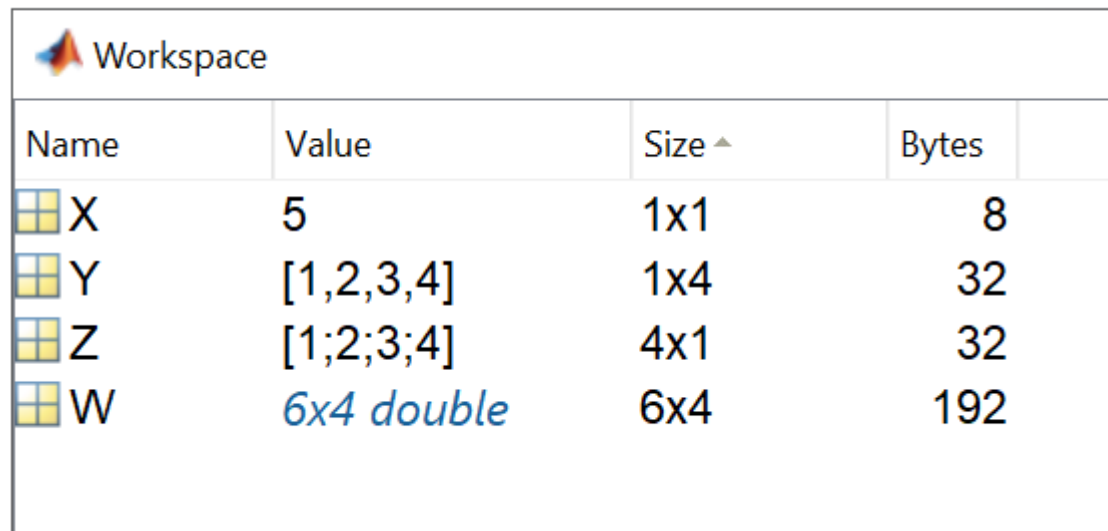


# Variables in MATLAB

# 1st principal

All MATLAB variables are multidimensional *arrays*, no matter what type of data.



The image shows a screenshot of the MATLAB Workspace window. The window title is "Workspace" with the MATLAB logo. It contains a table with four columns: "Name", "Value", "Size", and "Bytes". The table lists four variables: X, Y, Z, and W. Variable X has a value of 5, size 1x1, and 8 bytes. Variable Y has a value of [1,2,3,4], size 1x4, and 32 bytes. Variable Z has a value of [1;2;3;4], size 4x1, and 32 bytes. Variable W has a value of 6x4 double, size 6x4, and 192 bytes.

Name	Value	Size	Bytes
X	5	1x1	8
Y	[1,2,3,4]	1x4	32
Z	[1;2;3;4]	4x1	32
W	6x4 double	6x4	192

two-dimensional arrays

# Array Creation

## row vector

- **By components listing**

```
>>a= [1 2 3 4]
```

```
>>a= [ 1 ,2 ,3 ,4]
```

```
a = 1 2 3 4
```

# Array Creation

## row vector

- **By components description**

```
>>a= 1:4
```

```
    a = 1 2 3 4
```

```
>>a= 1:2:8
```

```
    a = 1 3 5 7
```

```
>>a=1: -1: 9
```

```
    a=[] %empty
```

# Array Creation

## row vector

- **MATLAB functions**

linspace, logspace, rand, ones, zeros,

```
>> linspace(initial, final, cpt)
```

```
>> linspace(2, 9, 5)
```

```
ans =
```

```
2.0000 3.7500 5.5000 7.2500 9.0000
```

```
>> logspace(2, 9, 5)
```

```
ans =
```

```
100.00 5623.41 316227.77 17782794.10  
1000000000.00
```

# Array Creation

## row vector

- **Other methods :**

- **concatenation**

```
>>a=[1 2 3], b=[5 6 7],
```

```
>> c=[a, b]
```

```
c=
```

```
1 2 3 5 6 7
```

- **Import data**

MAT files : >>Load or File menu

# Array Creation

## column vector

- **By components listing**

```
>>a= [ 1 ; 2 ; 3 ; 4]
```

```
a=
  1
  2
  3
  4
```

- **By components description**

```
>> a=(1:4)'
```

```
>>a= ( 1:2:8)'      1
```

```
                a = 3
```

```
>>a=(1: -1: 9)'    5
```

```
                a=[] %empty 7
```

# Array Creation

## column vector

- **MATLAB functions**

```
>> linspace(2, 9, 5)'
```

```
ans =
```

```
2.0000
```

```
3.7500
```

```
5.5000
```

```
7.2500
```

```
9.0000
```

- **concatenation**

```
>> a=(1:4)'
```

```
>>b= ( 1:2:8)'
```

```
>>C=[a ; b]
```

- **Import data**

1

2

3

4

1

3

5

7



# Array Creation matrix

- **By components listing**

```
>>a= [ 1 2 3 ; 4 5 6 ; 7 8 9]
```

```
a=
```

```
    1    2    3  
    4    5    6  
    7    8    9
```

- **By components description**

```
a= [1 : 3 ; 4 : 6 ; 7 : 9]
```

```
b=[(1:3:7)', (2:3:8)', (3:3:9)']
```

# Array Creation matrix

- **MATLAB functions**

**>> function\_name(rows? , columns?)**

**>> ones(3,5)      >>rand(2,4)**

**>> zeros(3,5)      >>randi([-3,2], 2,5)**

## **Square matrices**

**>>ones(3)**

**>>zeros(4)**

**>>rand(2)**

**Can we create vectors using these functions?**

# Array Creation matrix

- **Concatenation**

convenient arrays size

```
>>C=[Z B; A D F]
```

```
>>cat(2,A,F)% [A,F].
```

```
>>cat(1,D,Z)% [D;Z].
```

variable	size
A	2X1
D	2X2
F	2X4
Z	3X2
B	3X5
C	?
ans	?
ans	?

# Array Creation practice

- Create a vector of odd values from 1 to 10
- Create a matrix of 4 rows by 4 columns of value 3
- Create a matrix of 2x5 random integer values ranged from 1 to 6

# Array Creation practice

```
>>A=1 : 2 : 10
```

```
>>M=3*ones(4)
```

```
>>Z=randi([1,6],2,5)
```