L19. Two-Dimensional Arrays

- Set-Up
- Rows and Columns
- Subscripting
- Operations
- Examples

Simple Set-Up Examples

```
>> A = [1 2 3; 4 5 6]
A =
     1     2     3
     4     5     6
```

Simple Set-Up Examples

```
>> A = zeros(3,4)
A =
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

Simple Set-Up Examples

```
>> A = floor(100*rand(5,5))
A =
    95    76    61    40     5
    23    45    79    93    35
    60     1    92    91    81
    48    82    73    41     0
    89    44    17    89    13
```

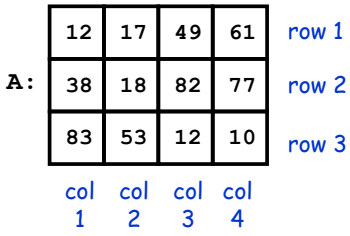
Simple Set-Up Examples

```
>> A = [zeros(3,2) [1;2;3]]
A =
     0     0     1
     0     0     2
     0     0     3
```

Simple Set-Up Examples

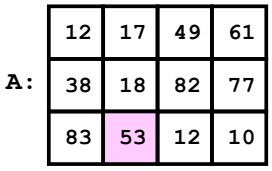
```
>> A = [zeros(3,2) ; [1 2] ]
A =
     0     0
     0     0
     0     0
     1     2
```

Rows and Columns



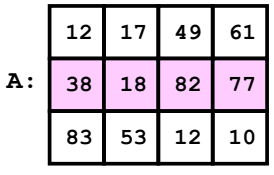
A is a 3-by-4 array: 3 rows 4 columns.

Subscripting



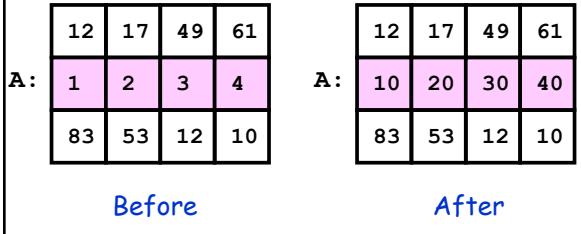
Individual entries: `A(3,2)`

Subscripting



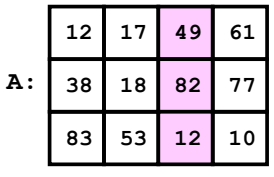
An Entire Row: `A(2,:)`

Scaling a Row



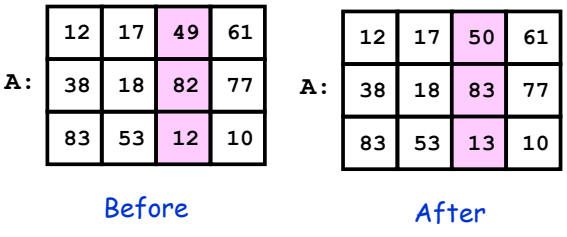
`A(2,:) = 10*A(2,:)`

Subscripting



An Entire Column: `A(:,3)`

Incrementing the Values in a Column



`A(:,3) = A(:,3) + 1`

Subscripting

A:

12	17	49	61
38	18	82	77
83	53	12	10

A General Subarray: `A(2:3,3:4)`

Zeroing a Subarray

A:

12	17	49	61
38	18	82	77
83	53	12	10

Before

A:

12	17	49	61
38	18	0	0
83	53	0	0

After

`A(2:3,3:4) = zeros(2,2)`

Classical Double Loop Set-Up

A:

11	21	31	41
12	22	32	42
13	23	33	43

```
for i=1:3
   for j=1:4
      A(i,j) = 10*j + i;
   end
end
```

Set-Up By Row

A:

11	21	31	41
12	22	32	42
13	23	33	43

```
A = [];
for i=1:3
   v = [10 20 30 40] + i;
   A = [A ; v]
end
```

Set-Up By Column

A:

11	21	31	41
12	22	32	42
13	23	33	43

```
A = [];
for j=1:4
   v = 10*j + [1;2;3];
   A = [A  v]
end
```

Largest Value

A:

12	17	49	61
38	18	82	77
83	53	12	10

m:

83	53	82	77

M: 83

m = max(A) ; M = max(m)

Functions and 2D Arrays

```
function  alpha = Ave(A)
% A is a 2D array.
% alpha is the average of its
% values.
```

```
10 20 30
40 50 60
```
-> (10+20+30+40+50+60)/6

Need Built-In Function `size`

```
function  alpha = Ave(A)
[m,n] = size(A);
```

Add up all the numbers in the array. Store in s.

```
alpha = s/(m*n);
```

`size(A)` returns #rows and # columns

Refine…

```
function  alpha = Ave(A)
  [m,n] = size(A);
  s = 0;
  for i=1:m
    sRow = the sum of the values in A(i,:)
    s = s + sRow
  end
  alpha = s/(m*n);
```

```
sRow = the sum of the
       values in A(i,:)
```

```
sRow = 0;
for j=1:n
  sRow = sRow + A(i,j);
end
```

```
function  alpha = Ave(A)
[m,n] = size(A);
s = 0;
for i=1:m
    sRow = 0;
    for j=1:n
      sRow = sRow + A(i,j);
    end
    s = s + sRow
end
alpha = s/(m*n);
```