

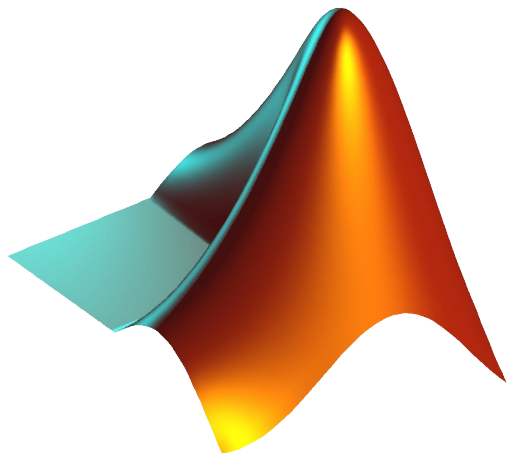
CS 1112 Introduction to Computing Using MATLAB

Instructor: Dominic Diaz

Website:

<https://www.cs.cornell.edu/courses/cs1112/2022fa/>

Today: cell arrays and file input



Agenda and announcements

- Last time
 - Finished char arrays
 - Cell arrays
- Today
 - More cell arrays
- Announcements
 - Project 4 due 10/27
 - Late deadline is 10/28 with a 5% penalty
 - Prelim 2 is 7:30 - 9 PM on Nov 10th in 305 Ives hall
 - Review session Nov 8th from 6:30 - 8 PM in Gates G01
 - If you have an SDS letter, check “prelim 2 time and location” on CMS for your time and location. If you have a university-approved conflict, submit a regrade request on this CMS assignment by November 1st.
 - Topics posted on the prelim 2 page of the website
 - Practice exams posted

| Course component | Percentage of grade |
|---------------------------|---------------------|
| Discussion exercises | 5% |
| Projects | 40% |
| Prelim 1 | 15% |
| Prelim 2 | 15% |
| Final | 24% |
| Poll Everywhere questions | 1% |

Data types we've seen so far

```
a = 45;
```

Basic (simple) data types

- **double** (stores any number 0.5, -45, 3.14, 0, ...)
 - MATLAB's default way to store any number
- **Boolean/logical** (stores true or false)
- **Uint8** (stores integers between 0 and 255, inclusive)
 - Usually used for images
- **Char** (stores a single character)

```
b = a > 0;
```

```
c = uint8(a);
```

```
d = 't';
```

Arrays

- **Numeric arrays**
 - 1D numeric array, 2D numeric array, and 3D numeric array. The numbers in these arrays could be of type double, logical, or uint8, but they all must be the same type.
- **Char arrays**
 - 1D char array, 2D char array
- **Cell arrays**
 - 1D cell array and 2D cell array

```
e = [1, 2, 3];
```

```
f = 'work';
```

```
g = {1, 'tr'};
```

Examples of arrays

Numeric array
of size 4x1

| |
|-----|
| 3.1 |
| 2 |
| -1 |
| 1.1 |

Char array of
size 4x3

| | | |
|-----|-----|-----|
| 'd' | 'a' | 'd' |
| 'm' | 'o' | 'm' |
| 'c' | 'a' | 't' |
| 'd' | 'o' | 'g' |

Cell array of
size 3x2

| | |
|----------|----|
| 'Daniel' | 18 |
| 'Luca' | 19 |
| 'Ashley' | 18 |

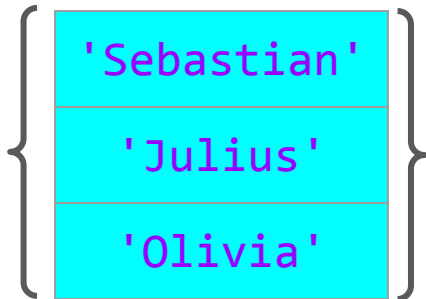
Application: cell array of 1D char arrays

```
C = {'Amanda', 'Matthew', 'Viviana'};
```



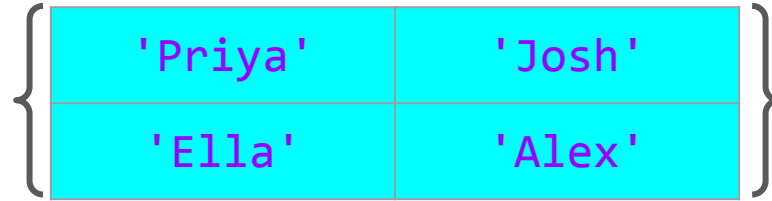
Commas create rows

```
C = {'Sebastian'; 'Julius'; 'Olivia'};
```



Semicolons create columns

```
C = {'Priya', 'Josh';  
     'Ella', 'Alex'};
```



Cell array

- The elements in a cell array can be of any type!

- A number (double)
- Boolean (logical) value
- Uint8 number
- Character (char)
- Any array
 - Including another cell array

| | | |
|------|-----------------|-----------------------|
| 10.5 | false | uint8(255) |
| '6' | [1, 2; 3, 4] | {'Spooky szn', 10} |

```
% Create the above cell array (method 1)
c = cell(2,3); % initialize cell array with 2
               % rows and 3 columns. Each cell
               % is initially empty.
```

```
c{1,1} = 10.5;
c{1,2} = false;
c{1,3} = uint8(255);
```

```
c{2,1} = '6';
c{2,2} = [1, 2; 3, 4];
c{2,3} = {'Spooky szn'; 10};
```

```
% Create the above cell array (method 2)
```

```
c = {10.5, false, uint8(255); '6', [1, 2; 3, 4], {'Spooky szn',10}};
```

Accessing parts of a cell array

| | | |
|------|-----------------|-----------------------|
| 10.5 | false | uint8(255) |
| '6' | [1, 2; 3, 4] | {'Spooky szn', 10} |

How can we store the uint8 value in a variable called a?

```
a = c{1,3};
```

How can we store the number 3 from the numeric array in a variable called b?

```
b = c{2,2}(2,1);
```

why? c{2,2} accesses [1, 2;
3, 4]

c{2,2}(2,1) accesses 3

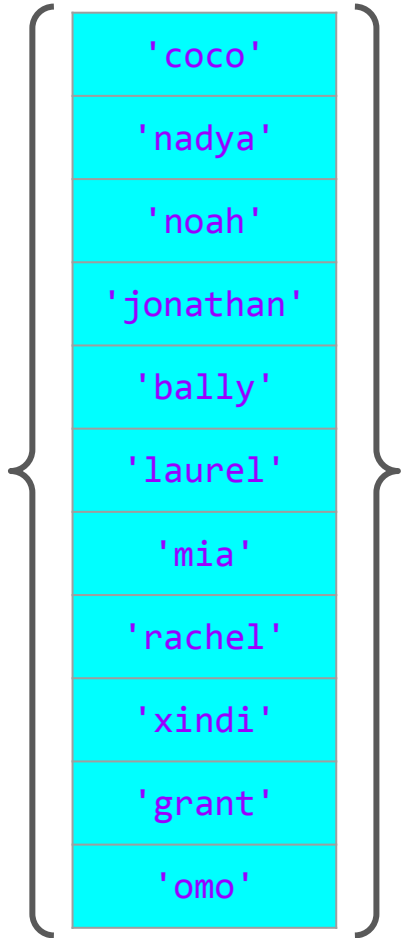
How can we store 'szn' from the char array into a variable called d?

```
d = c{2,3}{1}(8:10);
```

why? c{2,3} accesses {'Spooky szn', 10}

c{2,3}{1} accesses 'Spooky szn'

Example: processing cells in a cell array



A palindrome is a set of characters that reads the same backwards and forwards.

Examples: dad, mom, level, racecar, ...

Given a cell array `c` of size `n` by `1` with each cell storing a name, determine how many names are palindromes.

Pseudocode:

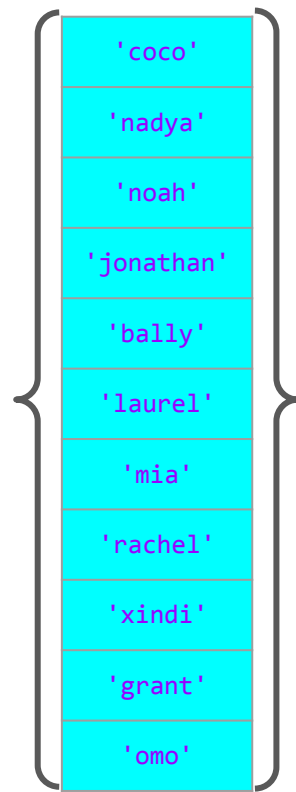
for each name

 reverse the name

 if reversed name = original name

 increment accumulator

```
function numPals = countPalin(names)
% Returns the number of palindromes in cell array names
% names is a nx1 cell array of char arrays
```



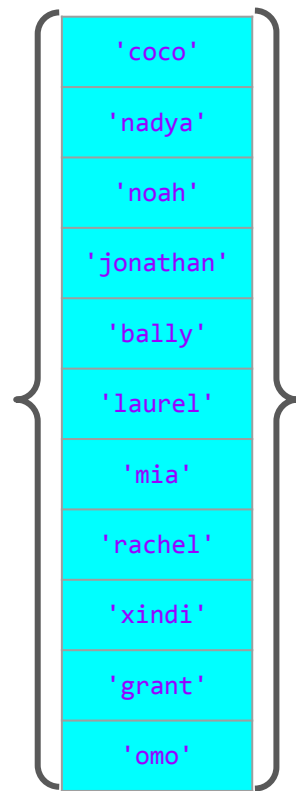
```
function numPals = countPalin(names)
% Returns the number of palindromes in cell array names
% names is a nx1 cell array of char arrays

numPals = 0;

[nr, nc] = size(names);
for i = 1:nr
    % store the current name and reverse it in nameReverse

    % Check if currName and nameReverse are the same

end
```



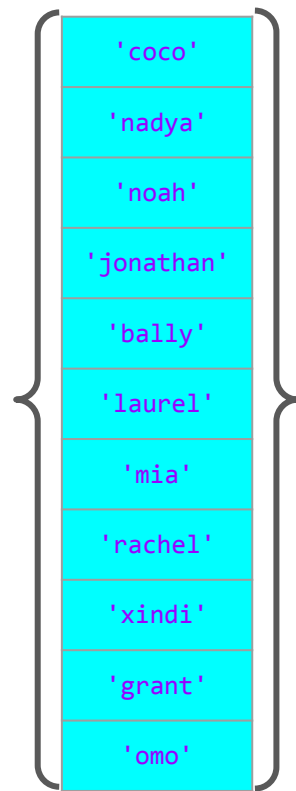
```
function numPals = countPalin(names)
% Returns the number of palindromes in cell array names
% names is a nx1 cell array of char arrays

numPals = 0;

[nr, nc] = size(names);
for i = 1:nr
    % store the current name and reverse it in nameReverse
    currName = names{i,1};

    % Check if currName and nameReverse are the same

end
```



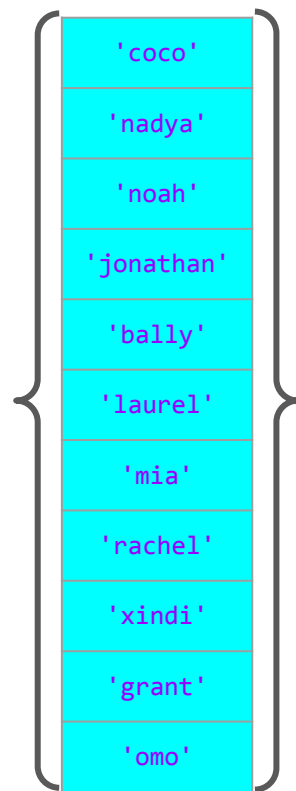
```
function numPals = countPalin(names)
% Returns the number of palindromes in cell array names
% names is a nx1 cell array of char arrays

numPals = 0;

[nr, nc] = size(names);
for i = 1:nr
    % store the current name and reverse it in nameReverse
    currName = names{i,1};
    nameReverse = '';
    for j = 1:length(currName)
        nameReverse(j) = currName(length(currName)-j+1);
    end
    % Check if currName and nameReverse are the same

end

end
```



```

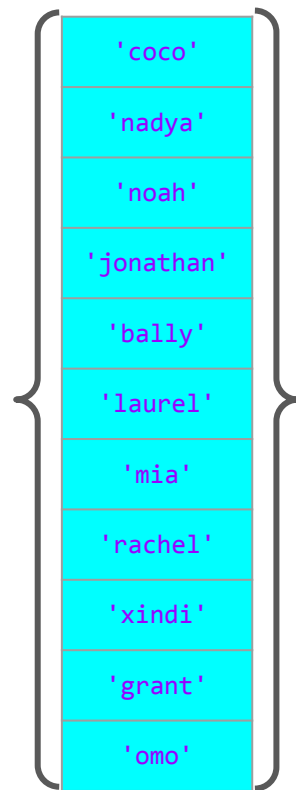
function numPals = countPalin(names)
% Returns the number of palindromes in cell array names
% names is a nx1 cell array of char arrays

numPals = 0;

[nr, nc] = size(names);
for i = 1:nr
    % store the current name and reverse it in nameReverse
    currName = names{i,1};
    nameReverse = '';
    for j = 1:length(currName)
        nameReverse(j) = currName(length(currName)-j+1);
    end
    % Check if currName and nameReverse are the same
    if strcmp(currName, nameReverse)
        numPals = numPals + 1;
    end
end

end

```



Working with text files

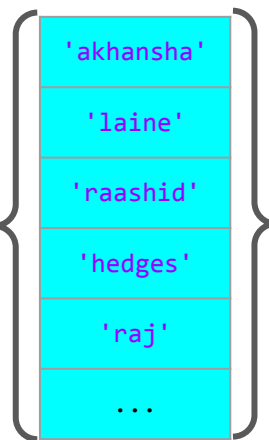
We've seen how to read an image file and store that in MATLAB:

```
img = imread('LawSchool.png');
```

But how can we read a text file and store that in MATLAB as a cell array?

names.txt

```
akhansha  
laine  
raashid  
hedges  
raj  
...
```



Read data from a file

1. **Open** a file
2. **Read** the file line-by-line until end-of-file
3. **Close** the file

function `fopen`

```
graph LR; A[1. Open a file] --> B[function fopen]; C[2. Read the file line-by-line until end-of-file] --> D[functions fgetl, feof]; E[3. Close the file] --> F[function fclose];
```

functions `fgetl, feof`

function `fclose`

1 & 3: open (and later close) the file

```
fid = fopen('names.txt', 'r');
```

An opened file has a file ID, here stored in a variable `fid`

Built-in function to open a file

Name of the file to open. `txt` and `dat` are common file name extensions for plain text files

'r' indicates that the file has been opened for reading.

Built-in function to close a file

```
fclose(fid);
```

Use semicolon after `fclose` because file commands return status codes

2: read each line and store it in a cell array

```
fid = fopen('names.txt', 'r');
```

```
k = 0;
```

```
C = {};
```

```
while ~feof(fid)
```

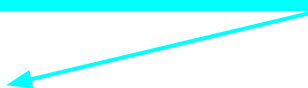
```
    k = k + 1;
```

```
    C{k,1} = fgetl(fid);
```

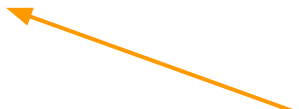
```
end
```

```
fclose(fid);
```

feof(fid) will be false until end-of-file is reached



Get the next line in file. (Each call gets one line; you cannot make it skip lines or go to a specific line.)



What is this code doing?

```
fid = fopen('names.txt', 'r');
```

```
k = 0;
```

```
C = {};
```

```
while ~feof(fid)
```

```
    k = k + 1;
```

```
    C{k,1} = fgetl(fid);
```

```
end
```

```
fclose(fid);
```

Opens names.txt

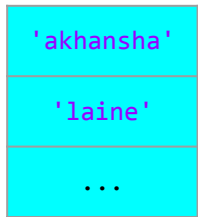
```
akhansha  
laine  
raashid  
hedges  
raj  
...
```



Creates C = {};



Loops through lines of file
and stores line in C.



```
function CA = file2Array(fname)
% returns an nx1 cell array of each line in fname
% input fname is a file name referring to the name of the txt file

fid = fopen(fname, 'r');
k = 0;
CA = {};
while ~feof(fid)
    k = k+1;
    CA{k,1} = fgetl(fid);
end
fclose(fid);
```

We would call this function with input
fname = 'names.txt';

Why type is fname?
Char array

Computing the number of people whose names are palindromes in CS 1112

```
C = file2cellArray('names.txt');  
numPals = countPalin(C);
```

```
function CA = file2Array(fname)  
% returns an nx1 cell array of each line in fname  
% input fname is a file name referring to the name of the txt file  
...
```

```
function numPals = countPalin(names)  
% Returns the number of palindromes in cell array names  
% names is a nx1 cell array of char arrays  
...
```