**Cell arrays** are a useful data type because **they can hold different types of data in each of their elements**. An element within a cell array is (surprise!) called a **cell**. In contrast to a double, which is created using normal brackets [ ], a cell array is created using curly brackets { }.

|  |  |
| --- | --- |
| For example, a 3x1 cell array holding a 1x2 double in its first cell, a 2x3 cell array in its second cell, and a 1x13 char array in its third cell would look like this: | C = {1x2 double}  {2x3 cell}  {‘Hello, world!’} |

You can build cell arrays the same way you build doubles, where **each element (ie, cell) is separated by a space or a comma.**

|  |  |
| --- | --- |
| {1,2,3,4,5,6,7,8,9}  or  {1 2 3 4 5 6 7 8 9} | Creates a 1x9 cell array, where each cell contains a 1x1 double representing the numbers 1 through 9 |
| {1:9} | Creates a 1x1 cell array containing a 1x9 double |
| {1:9,1:9,1:9}  or  {1:9 1:9 1:9} | Creates a 1x3 cell array, where each cell contains a 1x9 double |

|  |  |
| --- | --- |
| **Ex 1a.** Create a 1x3 cell array whose first element is your name, second element is your age, and third element is a 1x3 double containing your birthday. | **C{1,1} = ‘Alice Hsu’; C{1,2} = 22; C{1,4} = [2 13 1998]** |

Like doubles, you can **preallocate a cell array**. This is done using the built-in MATLAB function cell().

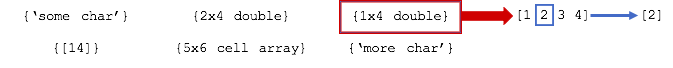
|  |  |  |
| --- | --- | --- |
| cell(n,m) | Creates an nxm cell array, where each cell is filled with an empty, 0x0 double | cell(2,3) =  {0x0 double} {0x0 double} {0x0 double}  {0x0 double} {0x0 double} {0x0 double} |

Indexing, accessing, and changing cells within a cell array are done the same way as a double, except **curly brackets { } are used** instead of parentheses ( ). For a given row **n** and column **m** in a cell array c\_array:

|  |  |
| --- | --- |
| To access a cell in a cell array | var = c\_array{n,m} |
| To change a cell in a cell array | c\_array{n,m}= some data |
| To access an entire row or column of cells of a cell array | cells = c\_array{n,:} or c\_array{:,m} |
| To change an entire row or column of cells of a cell array | c\_array{n,:} = data or c\_array{:,m} = data |

|  |  |  |
| --- | --- | --- |
| **Ex 1b.**  C =  {1x2 double} {2x3 cell}  {‘Hello’} {[12345]} | **Ex 2b. The cell array pet\_count tracks the number and kinds of pets I have. If I were to adopt 30 cats and 150 llamas, how would I update my cell array? (Place the llama count in the 4th column.)**  pet\_count =  {‘dog’} {‘cat’} {‘bird’} {0x0 double}  {[2]} {[0]} {[0]} {0x0 double} | |
| **C{1,2} = a 2x3 cell, whose contents we could see if accessed** | **pet\_count{2,2} = 30**  **pet\_count{1,4} = ‘llama’** | **pet\_count{2,4} = 150** |

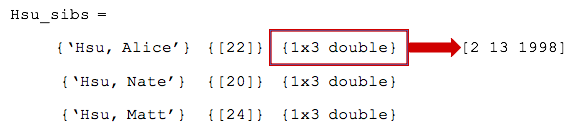
As stated before, an important difference between doubles and cell arrays is that **cell arrays can store different data types, including other arrays, in each of its cells.** We now know how to access the contents of a single cell within a cell array. But how would I **access data within an array within a cell within a cell array**?



To access data within an array that is being stored in a cell array, you must first specify the cell index using the curly brackets { }. Then you specify the element index within the array being stored in the cell using parentheses ( ). For a given row **i** and column **j** in an array being stored in a cell in row **n** and column **m** of c\_array:

c\_array{n,m}(i,j)

Suppose I have a 3x3 cell array called Hsu\_sibs containing the names, ages, and birthdays of my brothers and I in the first, second, and third columns, respectively.



|  |  |  |
| --- | --- | --- |
| **Ex 1c.** How would I extract my birth month from Hsu\_sibs? | **Ex 2c.** How would I extract Nate’s age from Hsu\_sibs? | **Ex 3c.** How would I extract my older brother’s first name from the char array in the first column? (Hint: what are the indices of the letters in his name within the char array?) |
| **Hsu\_sibs{1,3}(1)** | **Hsu\_sibs{2,2}** | **Hsu\_sibs{3,1}(6:9)** |