Single program, multiple data – spmd

The spmd statement defines a block of code to be run simultaneously on multiple workers that should be reserved using parpool.
The first couple of slides are a gentle adaptation of the text in:
https://se.mathworks.com/help/distcomp/spmd.html
The general form of an spmd (single program, multiple data) statement is:

```
spmd
    statements
end
```

MATLAB® executes the spmd body denoted by statements on several MATLAB workers simultaneously.

Open pool, `parpool`, `labindex`, `numlabs`

First open a pool of MATLAB workers using `parpool` or have your parallel prefences allow the automatic start of a pool. Inside the body of the spmd statement, each MATLAB worker has a unique value of `labindex`, while `numlabs` denotes the total number of workers executing the block in parallel. Within the body of the spmd statement, communication functions for communicating jobs (such as `labSend` and `labReceive`) can transfer data between the workers.

```
nlabs=2;   % laptop
% nlabs=16;  % Triton
parpool(nlabs)
spmd
  % build magic squares in parallel
  q = magic(labindex + 2);
  % for ii=1:length(q);figure,imagesc(q{ii});end
  % Works,as q{ii} brings it to the client.
end
```

The variable `q` is a Composite object. The value `q{k}` is the value stored in the k^{th} worker.

Composite objects

```
>> q
q =
   Lab 1: class = double, size = [3   3]
   Lab 2: class = double, size = [4   4]

>> q{1:nlabs}
ans =
     8     1     6
     3     5     7
     4     9     2
ans =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

Composite objects to cell arrays, results from workers to client

The variable q "behaves" like a cell array. After closing the pool it isn't available (the workers are gone). Here's how to bring q into the client:

```
Q=cell(1,nlabs);          % Create a cell arrays Q.
Q(1:nlabs)=q(1:nlabs);   % Copy q to cell array Q.
% Q=q;                    % Not allowed.
delete(gcp)    % q no more availeble, Q remains.
cellplot(Q), figure
imagesc(Q{nlabs})  % Plot the last (largest) magic.
```