SCISSORS parse MESSAGES

Concept:

Individual ELEMENTS of a PACKAGED MESSAGE can be parsed (extracted) using SCISSORS. MESSAGES must be packed with GLUE. ELEMENTS are extracted by *index* and can be *accessed out of order*. To retrieve an ELEMENT, you must know its index and type (integers, floats or Strings). Versions of SCISSORS are available for Processing and Arduino.

Arduino SCISSORS is a LIBRARY and must be in the *~/Documents/Arduino/libraries* folder (OSX).
Processing SCISSORS is a CLASS, just drop it on your open sketch. You can find a copy of SCISSORS in MPM503_code/utilityClasses/Scissors.pde

Code:

Processing SCISSORS:

```
// import
 // drop SCISSORS.pde onto SKETCH
   or SKETCH --> ADD FILE --> SCISSORS.pde

// declare
   Scissors scissors;

setup() {
 scissors = new Scissors( STREAM ); // instantiate object

     // where STREAM is the name of the SERIAL port, name of
     // the CLIENT or name of the SERVER which is receiving
     // messages.
}

draw() {

// poll the stream to see if anything has arrived

  if (scissors.update() > 0) {          // if new ELEMENTS

     float   f = scissors.getFloat(2);
     int     i = scissors.getInt(0);
     String s = scissors.getString(1);

  }

}
```

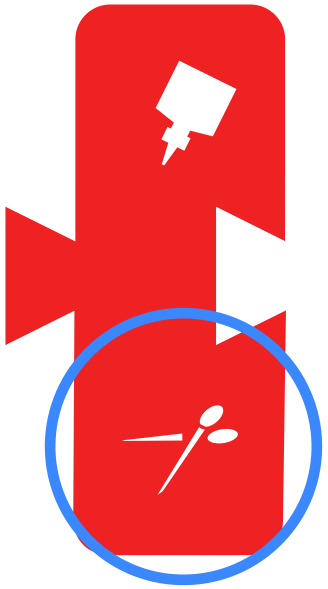
Arduino SCISSORS:

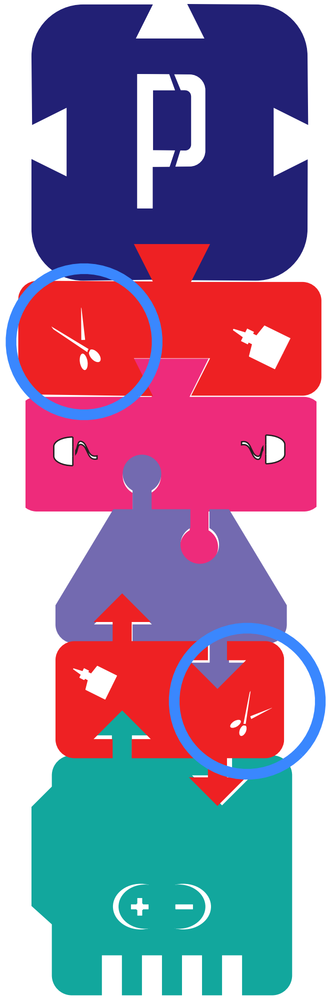
```
// import
#include <Scissors.h>

// declare
Scissors scissors;

setup() {
  scissors.begin(BAUD); // STREAM == SERIAL

    // must attach SCISSORS to a STREAM --> in ARDUINO
    // SCISSORS ALSO starts the STREAM.  As STREAM == the
    // SERIAL PORT -- you set BAUD when you begin SCISSORS
}

loop() {

// poll the stream to see if anything has arrived

  if (scissors.update() > 0) {        // if new ELEMENTS

     float    f = scissors.getFloat(2);
     int      i = scissors.getInt(0);
     String s = scissors.getString(1);

  }

}
```

COMPARE:
Arduino and Processing
handle this differently.

COMPARE:
Arduino and Processing
handle this differently.

Can retrieve out
of order.





Examples:

2_TCP_physical_network

Ver 1.0a