Decision Making
if and loops

- Need to do one of two things
  - if: do next instruction or skip to another (else)
  - loop: continue in loop or exit loop
Decision instructions in MIPS

- Only two such instructions:
  - \texttt{beq \$t0, \$t1, label}
    » If value in \$t0 equals value in \$t1, go to instruction at label
    » otherwise, execute next instruction in order
  - \texttt{bne \$t0, \$t1, label}
    » If value in \$t0 does not equal value in \$t1, go to instruction at label
    » otherwise, execute next instruction in order
if statement in assembly

if(a == b)  // a in $t0, b in $t1
    {c = 1}   // c in $s0
// next instruction
// next instruction

bne $t0, $t1, endif
addi $s0, $zero, 1
endif:  # next instruction
if - else

if (a == b)  // a in $t0, b in $t1
  {c = 1}  // c in $s0
else
  {c = 0}
// next instruction
if – else in assembly

bne $t0, $t1, else
addi $s0, $zero, 1
j endif

else: addi $s0, $zero, 0
endif: # next instruction
What about other comparisons?

- $a < b$
- $a <= b$
- etc...
Set less than instruction

- **slt $t0, $s1, $s2**
  - If value in $s1 is less than value in $s2, then set the value in $t0 to 1, otherwise set $t0 to 0

- Use slt combined with beq or bne, comparing to the $zero register, for an if comparison.
if(a < b) // a in $t0, b in $t1
        {c = 1}   // c in $s0

// next instruction

slt $t2, $t0, $t1
beq $t2, $zero, endif
addi $s0, $zero, 1
endif: # next instruction
while loop

while (condition)
{
    loop body
}

Each time at the top of the loop, check the condition. If true, continue the loop. At the end of the loop, go back to check the condition again.
while loop example - c

while (count < 10) {
    // do something
    count ++;
}

// next instruction
while loop example - assembly

# Assume count is in register $t0, # and register $s0 contains 10

loop:    slt $t1, $t0, $s0
         b?? $t1, $zero, loopend
         # do something
         addi $t0, $t0, 1
         j loop

loopepend:    # next instruction
while loop

# Assume count is in register $t0,
# and register $s0 contains 10

loop:     slt $t1, $t0, $s0
        beq $t1, $zero, loopend
        # do something
        addi $t0, $t0, 1
        j loop

loopend:
        # next instruction
for loop example - c

```c
for(cnt = 5, cnt >= 0, cnt--)
    {
        // do something
    }

// next instruction
```
for loop example - assembly

```assembly
addi $t0, $zero, 5

loop:
        slt $t1, $t0, $zero
        bne $t1, $zero, loopend

        # do something

        addi $t0, $t0, -1

        j loop

loopend:
        # next instruction
```