Chap 4 - SELECTION - DECISIONS

TWO TYPES

MOST POWERFUL
CAN DO
ANY DECISION →

1. "if" WITH ONE OR MORE OPTIONAL elseif
   WITH OPTIONAL else BEFORE end
   - CAN DECIDE WITH RANGES OF VALUES → if a < 5 && b > 2
     (a < 5 : RANGE; && : COMPOUND LOGICAL EXPRESSION.)
   - CAN DECIDE WITH SPECIFIC VALUES → if a == 5 && b == 2
     (== : EQUAL)

ONLY WITH
SPECIFIC VALUES →

2. "switch" WITH ONE OR MORE "case"
   WITH OPTIONAL "otherwise" before end.
   - CAN DECIDE WITH SPECIFIC VALUES (NOT RANGES)

- WHY USE SWITCH?
- WHEN APPLIES, FASTER TO TYPE & READ
- USED ESPECIALLY WITH MENUS

```
if condition 1
    do this 1 ——→ then go past end.
elseif cond 2
    do this 2. ——→ then go past end
elseif cond 3
    do this 3
else
    do this if none above are true
end
```

```
switch expression  → a, or a+b, etc.
    case aValueOfExpression
        do this ——→ then go past end.
    case value 2
        do this
    case { value 3, value 4, value 5 }
        do this
    otherwise
        do this if no cases above are true
end
```

Chap 5 – Loops, Repeats

Two Types

① "for" Loop

```
for i = 1:10
    actions
end
```

LOOP VARIABLE

RANGE OF VALUES
INTEGERS OR FLOATING PT.

actions → fprintf('i = %i \n', i)

- LOOP VARIABLE IS OFTEN AN INTEGER
  USED AS AN ARRAY INDEX IN ACTIONS

i = 1
i = 2
i = 3
⋮
i = 10

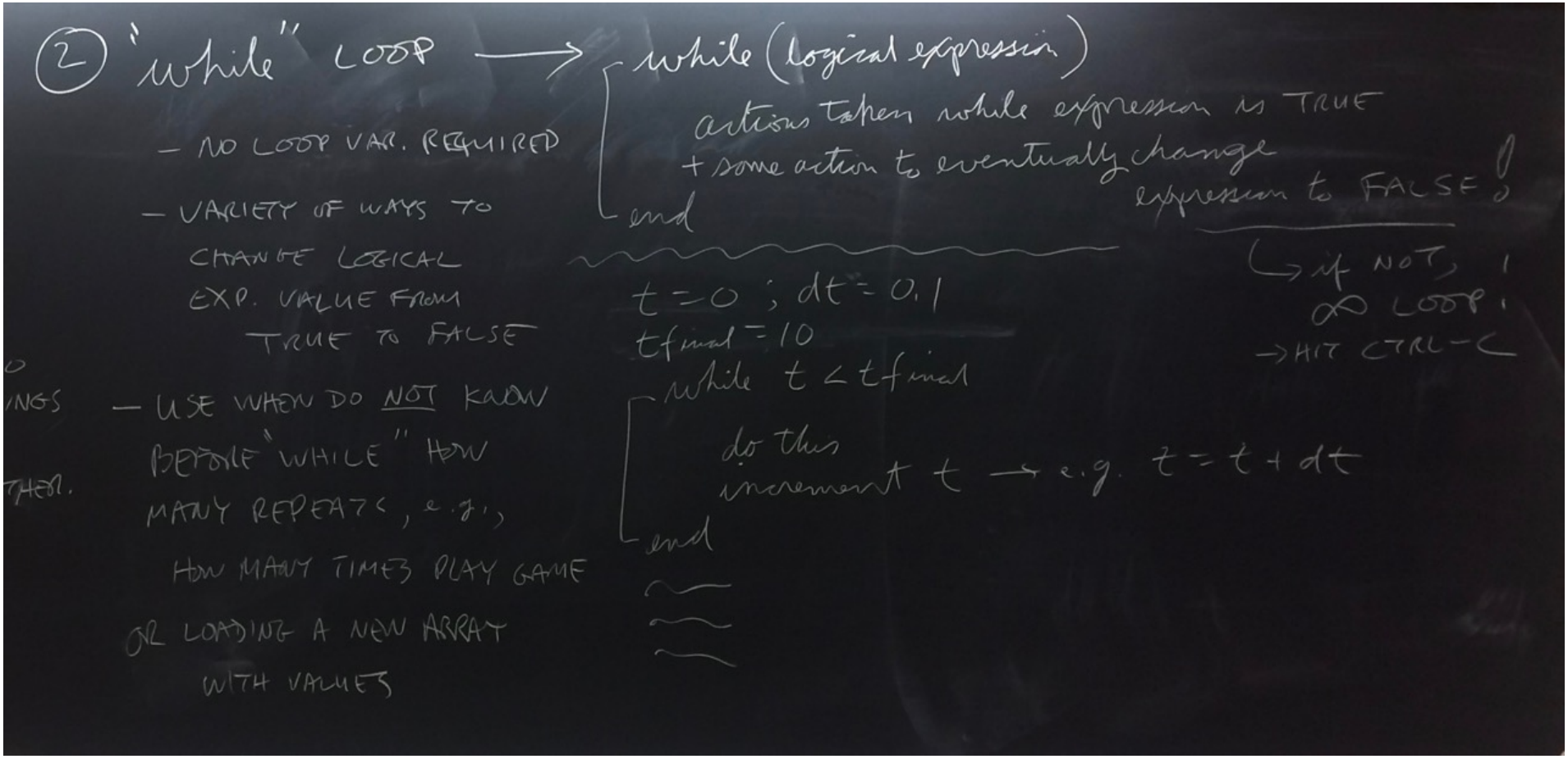
NOTE WE NEED TO SPECIFY RANGE OF VALUES OF LOOP VAR.

% S = ARRAY OF STUDENT ID'S

```
for i = 1:length(S)
    fprintf('student %i ID = %i \n', i, S(i))
end
```

- USE "FOR" WHEN KNOW BEFORE THE "FOR" HOW MANY TIMES TO REPEAT

- "FOR" ESPECIALLY USED TO PROCESS ARRAYS


(2) "while" LOOP
while (logical expression)
actions taken while expression is TRUE
+ some action to eventually change expression to FALSE!
end
- NO LOOP VAR. REQUIRED
- VARIETY OF WAYS TO CHANGE LOGICAL EXP. VALUE FROM TRUE TO FALSE
- USE WHEN DO NOT KNOW BEFORE "WHILE" HOW MANY REPEATS, e.g., HOW MANY TIMES PLAY GAME OR LOADING A NEW ARRAY WITH VALUES
t = 0 ; dt = 0.1
tfinal = 10
while t < tfinal
do this
increment t → e.g. t = t + dt
end
↳ if NOT, ∞ LOOP!
→ HIT CTRL-C

BASICS OF PROGRAMMING

(2)

INPUT, STORAGE, OUTPUT

PROCESS → e.g., $a = a + 2$

✓ DECISION

✓ REPEATS

FUNCTIONS

↓

A MAIN REASON COMPUTERS CAN DO SO MANY COMPLEX & AMAZING THINGS IS THAT REPEATS & DECISIONS CAN BE "NESTED" INSIDE EACH OTHER.