

Control structures

Kevin Schmidt, Susan Lindsey, Charlie Dey

Spring 2019

Conditionals

Conditionals

Single line conditional:

```
if ( test ) statement
```

The full if-statement is:

```
if ( something ) then
    do something
else
    do otherwise
end if
```

The 'else' part is optional; you can nest conditionals.

Comparison and logical operators

Operator	old style	meaning	example
==	.eq.	equals	$x == y - 1$
/=	.ne.	not equals	$x * x \neq 5$
>	.gt.	greater	$y > x - 1$
>=	.ge.	greater or equal	$\text{sqrt}(y) \geq 7$
<	.lt.	less than	
<=	.le.	less equal	
	.and. .or.	and, or	$x < 1 \text{ .and. } x > 0$
	.not.	not	$\text{.not.}(x > 1 \text{ .and. } x < 2)$
	.eqv.	equiv	$(x \wedge y) \vee (\neg x \wedge \neg y)$
	.neqv.	not equiv	$(x \wedge \neg y) \vee (\neg x \wedge y)$

Select statement

Test single values or ranges, integers or characters:

```
Select Case (i)
Case (:-1)
    print *, "Negative"
Case (5)
    print *, "Five!"
Case (0)
    print *, "Zero."
Case (1:4,6:) ! can not have (1:)
    print *, "Positive"
end Select
```

Compiler does checking on overlapping cases!