

# Introdução ao Fortran 90 - 2

Alexandre Diehl

Departamento de Física – UFPel

## Definição

**Funções Intrínsecas** são **subprogramas** disponibilizados pela linguagem **FORTRAN 90**, sem que o usuário precise defini-las. Podem ser usadas em expressões matemáticas ou em qualquer parte do programa.

## O que precisamos saber para usar uma **Função Intrínseca**

- O **nome** e o **significado** da Função.
- O **número de argumentos** usados pela Função.
- O **tipo** e as **restrições** de **cada argumento** usados pela Função.
- O **tipo do valor retornado** pela Função.

## Trigonométricas

Função	Significado	Restrições		Tipo de Argumento	
		Entrada	Saída	Entrada	Saída
<b>SIN (X)</b>	Seno do ângulo X	X em radianos	$-1 \leq \text{SIN}(X) \leq 1$	REAL	REAL
<b>COS (X)</b>	Cosseno do ângulo X	X em radianos	$-1 \leq \text{SIN}(X) \leq 1$	REAL	REAL
<b>TAN (X)</b>	Tangente do ângulo X	X em radianos	nenhuma	REAL	REAL
<b>ASIN (X)</b>	Arco seno de X	$ X  \leq 1$	$-\pi/2 \leq \text{ASIN}(X) \leq \pi/2$	REAL	REAL
<b>ACOS (X)</b>	Arco cosseno de X	$ X  \leq 1$	$0 \leq \text{ACOS}(X) \leq \pi$	REAL	REAL
<b>ATAN (X)</b>	Arco tangente de X	nenhuma	$-\pi/2 \leq \text{ATAN}(X) \leq \pi/2$	REAL	REAL
<b>ATAN2 (Y,X)</b>	Arco tangente para pares X e Y. Retorna o valor principal do número complexo (X,Y), expresso em radianos	X e Y não podem ser simultaneamente nulos	$-\pi \leq \text{ATAN2}(Y,X) \leq \pi$	REAL	REAL
<b>SINH (X)</b>	Seno hiperbólico de X	Nenhuma	nenhuma	REAL	REAL
<b>COSH (X)</b>	Cosseno hiperbólico de X	Nenhuma	nenhuma	REAL	REAL
<b>TANH (X)</b>	Tangente hiperbólica de X	Nenhuma	Nenhuma	REAL	REAL

```
real :: A, pi = 4.0*ATAN(1.0)
A = 3*COS(pi) - SIN(pi/2.0)
end
```

## Matemáticas comuns

Função	Significado	Restrições		Tipo de Argumento	
		Entrada	Saída	Entrada	Saída
ABS (X)	Valor absoluto de X	nenhuma	$ABS(X) \geq 0$	INTEGER	INTEGER
		nenhuma	$ABS(X) \geq 0$	REAL	REAL
SQRT (X)	Raiz quadrada de X	$X > 0$	nenhuma	REAL	REAL
		nenhuma	nenhuma	COMPLEX	COMPLEX
EXP (X)	Exponencial de X	nenhuma	nenhuma	REAL	REAL
		nenhuma	nenhuma	COMPLEXO	COMPLEXO
LOG (X)	Logaritmo natural de X	$X > 0$	nenhuma	REAL	REAL
		$X > 0$	$-\pi < \text{parte imaginária de } X < \pi$	COMPLEXO	COMPLEXO
LOG10 (X)	Logaritmo de base 10 de X	$X > 0$	nenhuma	REAL	REAL

```
real(8), parameter :: pi = 4.0*ATAN(1.0d0)
real(8) :: A, B
B = -10.0d0
A = 5.d0*ABS(B+3.0d0) + 1.0d0/(EXP(B*pi))
end
```

## Funções de Conversão

Função	Significado	Restrições		Tipo de Argumento	
		Entrada	Saída	Entrada	Saída
INT (X)	Converte X ao tipo inteiro	nenhuma	INT(X) = X	INTEGER	INTEGER
		nenhuma	valor truncado em sentido ao zero	REAL	INTEGER
		nenhuma	Somente a parte real de X tem valor truncado em sentido ao zero	COMPLEX	INTEGER
NINT (X)	Retorna o valor inteiro mais próximo de X	nenhuma	nenhuma	REAL	INTEGER
AINT (X)	Trunca o valor de X em sentido ao zero	nenhuma	Os valores decimais de X são igualados a zero	REAL	REAL
ANINT (X)	Retorna o valor real mais próximo de X, sem a parte fracionária	nenhuma	Os valores decimais de X são igualados a zero	REAL	REAL

```
real :: A = -3.65, B = 3.65, E, F
integer :: C, D
C = INT(A); D = NINT(A)
E = AINT(B); F = ANINT(B)
end
```

Resultado:

→ C = -3    D = -4    E = 3.00000000    F = 4.00000000

## Funções de Conversão

Função	Significado	Restrições		Tipo de Argumento	
		Entrada	Saída	Entrada	Saída
REAL (X)	Converte X ao tipo real	nenhuma	nenhuma	INTEGER	REAL
		nenhuma	nenhuma	REAL	REAL
		nenhuma	parte imaginária de X é ignorada	COMPLEX	REAL
FLOOR (X)	Retorna o <b>maior inteiro</b> menor ou igual a X	nenhuma	nenhuma	REAL	INTEGER
CEILING (X)	Retorna o <b>menor inteiro</b> maior ou igual a X	nenhuma	nenhuma	REAL	INTEGER
AIMAG (X)	Retorna a <b>parte imaginária</b> do complexo X	nenhuma	nenhuma	COMPLEX	REAL
CMPLX (X)	Converte X ao tipo complexo	nenhuma	nenhuma	INTEGER	COMPLEX
		nenhuma	nenhuma	REAL	COMPLEX
		nenhuma	nenhuma	COMPLEX	COMPLEX
CMPLX (X,Y)	Converte (X,Y) ao tipo complexo	nenhuma	nenhuma	INTEGER	COMPLEX
		nenhuma	nenhuma	REAL	COMPLEX

```
real :: a = 4.5; integer :: b = 4; complex :: c, d
c = CMPLX(a); d = CMPLX(b)
end
```

Resultado:

→ c = (4.50000000, 0.00000000)    d = (4.00000000, 0.00000000)

## Funções de Conversão

Função	Significado	Tipo de Argumento	
		Entrada	Saída
CONJG (X)	Retorna o conjugado de X	COMPLEX	COMPLEX
MAX (X1, X2, ..., Xn)	Retorna o máximo ( <b>maior valor</b> ) entre dois ou mais valores	INTEGER	INTEGER
		REAL	REAL
MIN (X1, X2, ..., Xn)	Retorna o mínimo ( <b>menor valor</b> ) entre dois ou mais valores	INTEGER	INTEGER
		REAL	REAL
MOD (X,Y)	Retorna o restante de X módulo Y: $X - \text{INT}(X/Y)*Y$ Se $Y = 0$ o resultado depende do processador	INTEGER	INTEGER
		REAL	REAL
MODULO (X,Y)	Retorna X módulo Y: $X - \text{FLOOR}(X/Y)*Y$ A divisão é a matemática ordinária. Se $Y = 0$ o resultado depende do processador	INTEGER	INTEGER
	Retorna X módulo Y: $X - \text{FLOOR}(X/Y)*Y$ Se $Y = 0$ o resultado depende do processador	REAL	REAL
SIGN (X,Y)	Retorna o valor absoluto de X vezes o sinal de Y. Se $Y = 0$ ( <b>INTEGER</b> ) o sinal é assumido positivo. Se $Y = 0.0$ ( <b>REAL</b> ) o sinal é assumido positivo	INTEGER	INTEGER
		REAL	REAL

```
real :: a, b
a = MAX(1.0, -4.5, 10.0); b = MIN(1.0, -4.5, 10.0)
end
```

Resultado: → a = 10.00000000 b = -4.50000000

## Funções de Conversão

Função	Significado	Tipo de Argumento	
		Entrada	Saída
CONJG (X)	Retorna o conjugado de X	COMPLEX	COMPLEX
MAX (X1, X2, ..., Xn)	Retorna o máximo ( <b>maior valor</b> ) entre dois ou mais valores	INTEGER	INTEGER
		REAL	REAL
MIN (X1, X2, ..., Xn)	Retorna o mínimo ( <b>menor valor</b> ) entre dois ou mais valores	INTEGER	INTEGER
		REAL	REAL
MOD (X,Y)	Retorna o restante de X módulo Y: $X - INT(X/Y)*Y$ Se $Y = 0$ o resultado depende do processador	INTEGER	INTEGER
		REAL	REAL
MODULO (X,Y)	Retorna X módulo Y: $X - FLOOR(X/Y)*Y$ A divisão é a matemática ordinária. Se $Y = 0$ o resultado depende do processador	INTEGER	INTEGER
	Retorna X módulo Y: $X - FLOOR(X/Y)*Y$ Se $Y = 0$ o resultado depende do processador	REAL	REAL
SIGN (X,Y)	Retorna o valor absoluto de X vezes o sinal de Y. Se $Y = 0$ ( <b>INTEGER</b> ) o sinal é assumido positivo. Se $Y = 0.0$ ( <b>REAL</b> ) o sinal é assumido positivo	INTEGER	INTEGER
		REAL	REAL

```
complex :: a = (1.0,1.0) , b
b = CONJG(a)
end
```

Resultado:  $\rightarrow b = (1.00000000, -1.00000000)$



## Entrada de Dados

- Os dados podem ser inseridos através do **teclado**
- Os dados podem ser inseridos através de arquivos (veremos depois)

## Usando o teclado

- Usamos o comando **READ\***, **A1**, **A2**, ..., **An**
  - os dados de entrada não são formatados (**formato livre**)
- ou usamos o comando **READ(\*,\*)****A1**, **A2**, ..., **An**
  - os dados de entrada não são formatados (**formato livre**)
  - este formato admite entrada de dados formatada (veremos depois)
- As variáveis **A1**, **A2**, ..., **An** devem ser definidas no programa. Não precisam ser do mesmo tipo.

## Entrada de Dados

- Os dados podem ser inseridos através do teclado

```
INTEGER :: idade
REAL :: peso
CHARACTER(LEN=20) :: nome
READ*, nome, idade, peso
!READ(*,*)nome, idade, peso ! forma alternativa
END
```

Digitar no teclado:

'Alexandre Diehl', 51, 87.0 → OK

'Alexandre Roberto Diehl', 51, 87 → nome com mais de 20 caracteres

## Saída de Dados

### Saída de Dados

- Os dados podem ser disponibilizados através do **monitor**
- Os dados podem ser disponibilizados através de arquivos (veremos depois)

### Usando o monitor

- Usamos o comando **PRINT\*, A1, A2, ..., An**
  - os dados de saída não são formatados (**formato livre**)
  - este formato admite entrada de dados formatada (veremos depois)
- ou usamos o comando **WRITE(\*,\*)A1, A2, ..., An**
  - os dados de saída não são formatados (**formato livre**)
  - este formato admite saída de dados formatada (veremos depois)
- As variáveis **A1, A2, ..., An** devem ser definidas no programa. Não precisam ser do mesmo tipo.

## Saída de Dados

- Os dados podem ser disponibilizados através do **monitor**

```
INTEGER :: idade  
REAL :: peso  
CHARACTER(LEN=20) :: nome  
READ(*,*) nome, idade, peso  
PRINT*,nome, idade, peso  
END
```

Digitar no teclado:

'Alexandre Diehl', 51, 87.0 → OK

Saída no monitor:

Alexandre Diehl 51 87.00000000

## Saída de Dados

- Os dados podem ser disponibilizados através do **monitor**

```
INTEGER :: idade
REAL :: peso
CHARACTER(LEN=20) :: nome
READ(*,*) nome
READ*, idade
READ(*,*) peso
PRINT*, 'nome=', nome
WRITE(*,*) 'idade=', idade
PRINT*, 'peso=', peso
END
```

- Comentários (caracteres) são inseridos entre **".."** ou **'..'**

## Saída de Dados

- Os dados podem ser disponibilizados através do **monitor**

```
real :: theta
read *, theta
print *, 'valor_do_angulo=', theta
print *, 'valor_do_seno_do_angulo=', SIN(theta)
write (*,*) 'tangente=', SIN(theta)/COS(theta)
END
```

- Podem ser usadas **expressões aritméticas**
- Podem ser usadas **Funções Intrínsecas**