PUNTEROS

- **Declaración** : tamaño *p;
  tamaño = el tamaño de la variable a la que apunta
  Ejemplo:
  short *p = &a; => la variable a la que apunta p (a) es un short o un vector de shorts
  Traducción a MIPS => p : .word a
  Como la variable p contiene una dirección de memoria (a la que apunta) siempre ocupa 1 word (4 bytes)

- **Cuando el puntero es una variable global**
  Sintaxis (a la hora de usarlo, no de declararlo):
  - *p : contenido de la dirección a la que apunta. En el ejemplo de antes sería el contenido de la variable a
  - p : contenido de p (dirección a la que apunta). Sería la dirección de a (&a)
  - &p : dirección de memoria en la que esta almacenada p

  Cómo conseguir el valor de cada una:

```
la $t0, p          #$t0 = &p
lw $t0, 0($t0)  #$t0 = p
lh $t0, 0($t0)   #&t0 = *p     *es un lh ya que la variable a la que apunta es un short*
```

- **Cuando el puntero es una variable local**
  Cuando el puntero es una variable local no está almacenado en memoria. Por lo tanto la dirección a la que apunta está guardada en un registro
  Sintaxis:
  - *p : contenido de la dirección a la que apunta
  - p : contenido de p (dirección a la que apunta)
  - ***NO EXISTE &p ya que p no está almacenada en memoria***

  Cómo conseguir el valor de cada una:
  ya no necesitamos hacer el *la $t0, p* y el *lw $t0, 0($t0)* ya que la dirección a la que queremos acceder ya la tenemos en un registro. Digamos que p está almacenada en el registro $t0. Por lo tanto:

```
                    #$t0 = p
lh $t0, 0($t0)   #$t0 = *p (a)
```

- **Iteración con punteros**
  Cuando el puntero apunta a un vector:
  - en C: si queremos que el puntero p apunte al siguiente elemento del vector
    p = p +1; *o* ++p;
  - en MIPS: digamos que p está apuntando a un vector de shorts (V), al elemento V[1]

  al principio p contiene la dirección de V[1]. Si queremos que apunte a V[2], entonces tendremos que sumarle 2, ya que cada elemento del vector ocupa 2 posiciones de memoria. De esta forma si por ejemplo V[1] está almacenada en 0x10010002, al sumarle dos a esta dirección tendremos en p la dirección 0x10010004, que es donde está alamacenada V[2]