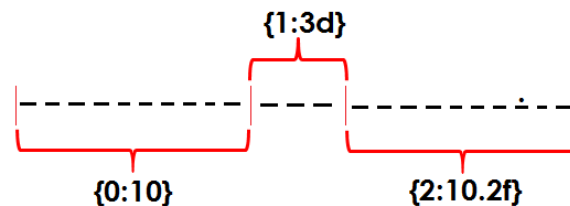


Formato de salida en Python

Para dar formato de salida a datos generados por un programa Python usaremos este ejemplo: `print("{0:10} {1:3d} {2:10.2f}".format(nombre, edad, sueldo))`, como vemos la instrucción de formato está formada por dos partes, la primera la cadena de formato (texto en verde) y separado por un punto la función `format` la cual entre paréntesis se lista las variables o constantes a los cuales se les aplicara el formato definido en la cadena de formato, en este ejemplo encerrados por llaves hemos definido tres tipos de formato para tres tipos de valores diferentes, analicemos el primero, `{0:10}`, después de la llave colocamos un cero, este valor indica la posición del dato en la lista de variables o constantes definidos en `format` al que se le aplicara el formato, en este caso se aplica a la data guardada en la variable `nombre`, seguido se colocan dos puntos, (:), a continuación colocamos un valor entero, 10, este valor indica que reservamos una columna que tendrá como ancho diez caracteres en el cual acomodaremos un nombre, si este nombre excede en tamaño el ancho de esta columna, se pierde el formato, y la alineación de la tabla, el segundo formato definido es `{1:3d}`, el número 1 indica que se aplicara al siguiente dato que ocupa esta posición en la lista de variables o constante que aparece en la función `format`, en este caso a `edad`, después de los dos puntos se colocó `3d`, el tres indica el ancho de la columna donde se acomodara el valor y la `d` que el formato se aplicara a un valor entero, si entero excede los tres caracteres se pierde el formato y por último el siguiente formato definido es `{2:10.2f}`, el 2 indica que el formato se aplicara a la variable o constante que ocupa esta posición en la lista de valores definida en la función `format`, en esta caso a la data guardada en `sueldo`, después de los dos puntos, observamos lo siguiente: `10.2f`, el diez permite reservar una columna de un ancho de diez caracteres donde se colocara la data del sueldo, separado por un punto hemos colocado un 2, lo cual significa que de los diez caracteres reservados dos se reservan para colocar dos dígitos decimales, y finalmente la `f` indica que este formato se usara para mostrar valores reales, el punto decimal ocupa un espacio dentro de los diez caracteres definidos. Gráficamente tendríamos esto:



Asumamos este formato de salida:

```
print("{0:12} {1:7.2f} {2:12}".format(nom, prom, estatus))
```

Que nos indica este formato `{0:12}`, se imprimirán cadenas de caracteres en una columna de ancho 12, `{1:7.2f}`, se imprimirán valores reales en una columna de ancho 7 de los cuales un espacio se reserva para el punto decimal y dos para dos dígitos decimales, `{2:12}`, se imprimirán cadenas de caracteres en una columna de ancho 12, que se mostraran en los distintos formatos definidos, de acuerdo a la lista que tiene la función `format`, un nombre, un promedio y un estatus, es de hacer notar que entre cada par de llaves se dejaron espacio en blanco adicionales, esto con la finalidad de dar más separación entre las columnas de formato, quedando la lista de la siguiente manera:

Nombre	Nota Final	Estatus
Juan	11.67	APROBO
Maria	8.33	REPROBO
Jose	17.00	APROBO
Nora	17.00	APROBO
Frank	11.00	APROBO
Rosa	19.00	APROBO
Rita	6.00	REPROBO
Roberto	10.33	APROBO
Gonzalo	10.33	APROBO
Alirio	8.00	REPROBO
Margarita	7.67	REPROBO
Luis	12.33	APROBO
Oswaldo	14.33	APROBO