

**UNIVERSIDAD DR. ANDRES BELLO
INGENIERIA EN SISTEMAS Y
COMPUTACION**

**FUNDAMENTOS DE
PROGRAMACIÓN**

ESTUDIANTES:

JOSE NICOLAS ABREGO SALGUERO

JOSUE AARON CARTILLO

VALDIVIEZO

BRYAN EDENILSON QUINTANILLA

ALBERTO

GERARDO ANTONIO ROSA

MENJIVAR

ROCIO MARBELLY MORENO ERAZO

ROLANDO ALEXANDER PEREZ

LOPEZ

DOCENTE:

ING. JONATHAN FRANCISCO

CARBALLO.

CHALATENANGO, EL SALVADOR

2023.

DIFERENCIAS ENTRE PYTHON 2 Y PYTHON 3:

SINTAXIS:

Python 2: La impresión es una declaración (print) sin paréntesis.

Python 3: La impresión es una función (print()) que requiere paréntesis.

DIVISIÓN:

Python 2: La división de dos enteros con el operador / devuelve una división entera truncada.

Python 3: El operador / siempre devuelve un número decimal, y la división entera se realiza con el operador //.

UNICODE:

Python 2: Tiene dos tipos de datos para cadenas de texto: str y unicode.

Python 3: Eliminó el tipo unicode y convirtió str en una secuencia de caracteres Unicode por defecto.

GESTIÓN DE BYTES Y TEXTO:

Python 2: Trata cadenas de texto y bytes de manera similar.

Python 3: Introdujo los tipos str para texto y bytes para secuencias de bytes, mejorando la claridad y evitando errores relacionados con la codificación.

ITERACIÓN:

Python 2: zip() devuelve una lista.

Python 3: zip() devuelve un iterador, ahorrando memoria en conjuntos de datos grandes.

GESTIÓN DE EXCEPCIONES:

Python 2: Sintaxis generalizada (except Exception, e:) para capturar excepciones.

Python 3: Sintaxis más simple (except Exception as e:) para capturar excepciones.

FUNCIONES RANGE() Y XRANGE():

Python 2: Existen dos funciones para generar rangos: range() y xrange().

Python 3: Eliminó xrange() y range() ahora funciona como xrange() en Python 2, devolviendo un objeto rango en lugar de una lista.

IMPORTACIONES:

Python 2: Importaciones relativas confusas.

Python 3: Sintaxis clara y consistente para importaciones relativas.

CORRECCIONES DE ERRORES EN PYTHON 3:

- Mejor manejo de la codificación Unicode y caracteres especiales.
- Solución de problemas relacionados con la división entera y flotante.
- Mejor manejo de excepciones y sintaxis más clara.
- Separación clara entre cadenas de texto y secuencias de bytes.
- Mayor eficiencia en el uso de memoria al iterar sobre grandes conjuntos de datos.
- Mejor manejo de importaciones relativas para proyectos más grandes y complejos.