

Updated_Documentation

October 1, 2025

0.1 Imported Modules Documentation

0.1.1 `os` Module

- **`remove(path: str) -> None`**
Deletes a file at the specified path.
 - **`sep: str`**
The character used by the operating system to separate pathname components (e.g., `'/'` on Unix, `'\\'` on Windows).
-

0.1.2 `secrets` Module

- **`choice(sequence) -> item`**
Return a randomly chosen element from a non-empty sequence.
 - **`randbits(k: int) -> int`**
Return an integer with `k` random bits (cryptographically secure).
-

0.1.3 `gmpy2` Module

- **`is_prime(n) -> bool`** (imported as `isprime`)
Determines if a number is a probable prime. Returns `True` if `n` is likely prime.
-

0.1.4 `sys` Module

- Provides access to system-specific parameters and functions.
 - Here it is used for `sys.set_int_max_str_digits()` to handle very large integers safely.
-

0.1.5 `cv2` (OpenCV) Module

- Used for **camera operations** and image capture.
- Provides access to video devices and image saving functions.

0.1.6 time Module

- **sleep(seconds: float) -> None**
Suspends execution for the given number of seconds.
Used to give the camera time to initialize before capturing images.

1 captureImage() Function Documentation

1.1 Overview

Captures one or more images from a connected camera and saves them as JPEG files.

1.2 Parameters

- **path (str):** Directory where images will be saved.
- **total_images (int, optional):** Number of images to capture (default: 1).
- **camera (int, optional):** Camera index to use (default: 0).

1.3 Returns

- **list[str]:** List of file paths of the saved images.

1.4 Raises

- **Exception:** If the camera cannot be opened or frames cannot be captured.

2 NoiseRandom Class Documentation

2.1 Overview

NoiseRandom generates randomness by capturing images from cameras and extracting entropy. The images are processed and scrambled to create unpredictable random numbers, bytes, and primes.

2.2 Constructor

```
NoiseRandom(path: str, strength: int = 1, cameras: list[int] = [0],  
            disable_scramble: bool = False, disable_delete_images: bool = False)
```

2.2.1 Parameters

- **path (str):** Directory where captured images will be stored.
- **strength (int, default=1):** Number of images to capture per entropy pool. Must be 1.

- **cameras** (list[int], default=[0]): List of camera indices to randomly choose from.
 - **disable_scramble** (bool, default=False): If True, disables the scrambling step.
 - **disable_delete_images** (bool, default=False): If True, captured images are not deleted.
-

2.3 Public Methods

2.3.1 randomInt(get_bytes: bool = False) -> int | bytes

Generates a random integer (or raw bytes if `get_bytes=True`) from image entropy.

2.3.2 randomBytes(total_bytes: int, get_bytes: bool = False) -> int | bytes

Generates a random sequence of bytes of length `total_bytes`.

Returns either bytes or an int depending on `get_bytes`.

- **Raises:** ValueError if `total_bytes <= 0`.

2.3.3 randomPrime(total_bytes: int) -> int

Generates a random prime number of size `total_bytes`.

Keeps retrying until a prime is found.

- **Raises:** ValueError if `total_bytes <= 0`.

2.3.4 Fixed-Size Random Generators

- **random1024()** → int: Returns a random 1024-bit integer.
- **random2048()** → int: Returns a random 2048-bit integer.
- **random4096()** → int: Returns a random 4096-bit integer.

2.3.5 Fixed-Size Prime Generators

- **randomPrime1024()** → int: Returns a random 1024-bit prime.
 - **randomPrime2048()** → int: Returns a random 2048-bit prime.
 - **randomPrime4096()** → int: Returns a random 4096-bit prime.
-

2.4 Private Methods (Internal Use)

- **__deleteImages()**: Deletes captured images (unless disabled).
- **__captureImages()**: Captures images using the `captureImage()` function.

- `__scramble(data)`: Scrambles image byte data to maximize entropy.

3 Usage Examples

```
from noiserandom import NoiseRandom

# Initialize entropy generator
nr = NoiseRandom(path="images", strength=2, cameras=[0,1])

# Generate a random integer
rand_int = nr.randomInt()
print("Random Integer:", rand_int)

# Generate 32 random bytes
rand_bytes = nr.randomBytes(32, get_bytes=True)
print("Random Bytes:", rand_bytes)

# Generate a 1024-bit prime number
prime_1024 = nr.randomPrime1024()
print("Random 1024-bit Prime:", prime_1024)

# Generate a 4096-bit random integer
rand_4096 = nr.random4096()
print("4096-bit Random Integer:", rand_4096)
```