NumPy Mathematical Functions

1 Basic Mathematical Functions

- `numpy.add(a, b)`: Element-wise addition.

```python
import numpy as np

a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
result = np.add(a, b)  # result = [5, 7, 9]
```

- `numpy.subtract(a, b)`: Element-wise subtraction.

```python
result = np.subtract(a, b)  # result = [-3, -3, -3]
```

- `numpy.multiply(a, b)`: Element-wise multiplication.

```python
result = np.multiply(a, b)  # result = [4, 10, 18]
```

- `numpy.divide(a, b)`: Element-wise division.

```python
result = np.divide(a, b)  # result = [0.25, 0.4, 0.5]
```

- `numpy.floor_divide(a, b)`: Element-wise floor division.

```python
result = np.floor_divide(a, b)  # result = [0, 0, 0]
```

- `numpy.power(a, b)`: Element-wise exponentiation.

```python
result = np.power(a, 2)  # result = [1, 4, 9]
```

- `numpy.remainder(a, b)`: Element-wise remainder.

```python
result = np.remainder(a, 2)  # result = [1, 0, 1]
```

2 Trigonometric Functions

- `numpy.sin(x)`: Sine function.

```python
x = np.array([0, np.pi/2, np.pi])
result = np.sin(x)  # result = [0.0, 1.0, 0.0]
```

- `numpy.cos(x)`: Cosine function.

```
result = np.cos(x)  # result = [1.0, 0.0, -1.0]
```

- `numpy.tan(x)`: Tangent function.

```
result = np.tan(x)  # result = [0.0, inf, 0.0]
```

- `numpy.arcsin(x)`: Inverse sine function.

```
result = np.arcsin(1)  # result =  /2
```

- `numpy.arccos(x)`: Inverse cosine function.

```
result = np.arccos(1)  # result = 0
```

- `numpy.arctan(x)`: Inverse tangent function.

```
result = np.arctan(1)  # result =  /4
```

3 Exponential and Logarithmic Functions

- `numpy.exp(x)`: Exponential function.

```
result = np.exp(1)  # result = e
```

- `numpy.log(x)`: Natural logarithm.

```
result = np.log(np.e)  # result = 1.0
```

- `numpy.log10(x)`: Base-10 logarithm.

```
result = np.log10(100)  # result = 2.0
```

- `numpy.log2(x)`: Base-2 logarithm.

```
result = np.log2(8)  # result = 3.0
```

- `numpy.log1p(x)`: Natural logarithm of (1 + x).

```
result = np.log1p(0)  # result = 0.0
```

4 Hyperbolic Functions

- `numpy.sinh(x)`: Hyperbolic sine function.

```
result = np.sinh(0)  # result = 0.0
```

- `numpy.cosh(x)`: Hyperbolic cosine function.

```
result = np.cosh(0)  # result = 1.0
```

- `numpy.tanh(x)`: Hyperbolic tangent function.

```
result = np.tanh(0)  # result = 0.0
```

5 Rounding Functions

- `numpy.floor(x)`: Round down to the nearest integer.

```
result = np.floor(2.7)  # result = 2.0
```

- `numpy.ceil(x)`: Round up to the nearest integer.

```
result = np.ceil(2.3)  # result = 3.0
```

- `numpy.rint(x)`: Round to the nearest integer.

```
result = np.rint(2.5)  # result = 2.0
```

6 Other Mathematical Functions

- `numpy.abs(x)`: Absolute value.

```
result = np.abs(-5)  # result = 5
```

- `numpy.sum(x)`: Sum of array elements.

```
result = np.sum([1, 2, 3])  # result = 6
```

- `numpy.prod(x)`: Product of array elements.

```
result = np.prod([1, 2, 3])  # result = 6
```

- `numpy.cumsum(x)`: Cumulative sum.

```
result = np.cumsum([1, 2, 3])  # result = [1, 3, 6]
```

- `numpy.cumprod(x)`: Cumulative product.

```
result = np.cumprod([1, 2, 3])  # result = [1, 2, 6]
```

7 Additional Functions

- `numpy.sqrt(x)`: Square root.

```
result = np.sqrt(4)  # result = 2.0
```

- `numpy.square(x)`: Square of an array.

```
result = np.square([1, 2, 3])  # result = [1, 4, 9]
```

- `numpy.cbrt(x)`: Cube root.

```
result = np.cbrt(8)  # result = 2.0
```