Handling image coordinates

astropy.wcs

Translate coordinates between images and the sky

Key concepts: WCS objects

Initialization

Pixels <-> sky coords

astropy.wcs

- WCS = World Coordinate System
- projections of celestial sphere onto a plane
- convert between image and sky coordinates
- footprint of image on sky

WCS objects

Initialization

Pixels to sky coords

astropy.wcs

WCS objects

Initialization

Pixels to sky coords

- WCS constructor

```
>>> from astropy.wcs import WCS
```

- Initialize from an image header (FITS or text)

```
>>> from astropy.io import fits
>>> w = WCS(fits.getheader('data/w5.fits')
```

- Create from scratch

astropy.wcs

WCS objects

Initialization

Pixels to sky coords

- Convert pixel coordinates to celestial coords

```
>>> from astropy.io import fits
>>> w.pixel_to_world(10.5, 200)
```

- Convert celestial coordinates to pixels

```
>>> from astropy.coordinates import SkyCoord
>>> c1 = SkyCoord(ra=46.34604634,
          dec=59.44975255, unit='deg')
>>> w.world_to_pixel(c1)
```

- High-level API uses Python/C pixel convention

Displaying images with WCS

astropy.visualization

Transform image values, for visualization

Key concepts: Scale

Stretch

WCS axes

astropy.visualization

Scale

Stretch

WCS axes

- Scale: choose lower and upper image values to map to the interval [0:1]
- Percentile intervals
- Manual intervals
- Asymmetric Percentile intervals
- ZScale intervals

astropy.visualization

Scale

Stretch

WCS axes

- Stretch: map [0:1] to [0:1] linearly or non-linearly

- Linear
- Log
- Sqrt
- Squared
- Sinh
- Power
- PowerDist

astropy.visualization

Scale

Stretch

WCS axes

- Connects WCS object to matplotlib
- Labeled axes
- Coordinate grids
- Overlay markers with celestial coordinates