ekp_style

Documentation.

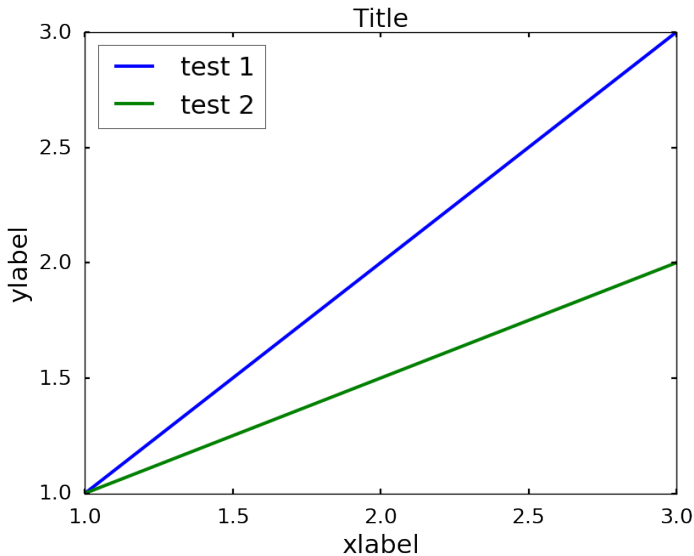Nils Braun | 26.02.2017

```
33      *
34      * What is more important can be controlled by the flag acceptOverridesReject, which is off by default (so reject i
35      * more important than accept by default).
36      */
37  class SoftwareTriggerModule : public Module {
38  public:
39      /// Create a new module instance and set the parameters.
40      SoftwareTriggerModule();
41
42      /// Initialize/Require the DB object pointers and any needed store arrays.
43      void initialize() override;
44
45      /// Run over all cuts and check them. If one of the cuts yields true, give a positive return value of the module.
46      void event() override;
47
48      /// Check if the cut representations in the database have changed and download newer ones if needed.
49      void beginRun() override;
50
51      /// Store and delete the ttree if it was created.
```

Basic style

```python
from ekp_style import set_ekp_style
set_ekp_style()

from matplotlib import pyplot as plt

plt.plot([1, 2, 3], [1, 2, 3], label="test 1")
plt.plot([1, 2, 3], [1, 1.5, 2], label="test 2")
plt.xlabel("xlabel")
plt.ylabel("ylabel")

plt.legend(loc=2, frameon=True)
plt.title("Title")
```

Basic style

Colored style

```python
from ekp_style import set_ekp_style
set_ekp_style(set_colors=True)

from matplotlib import pyplot as plt

plt.plot([1, 2, 3], [1, 2, 3], label="test 1")
plt.plot([1, 2, 3], [1, 1.5, 2], label="test 2")
plt.xlabel("xlabel")
plt.ylabel("ylabel")

plt.legend(loc=2, frameon=True)
plt.title("Title")
```
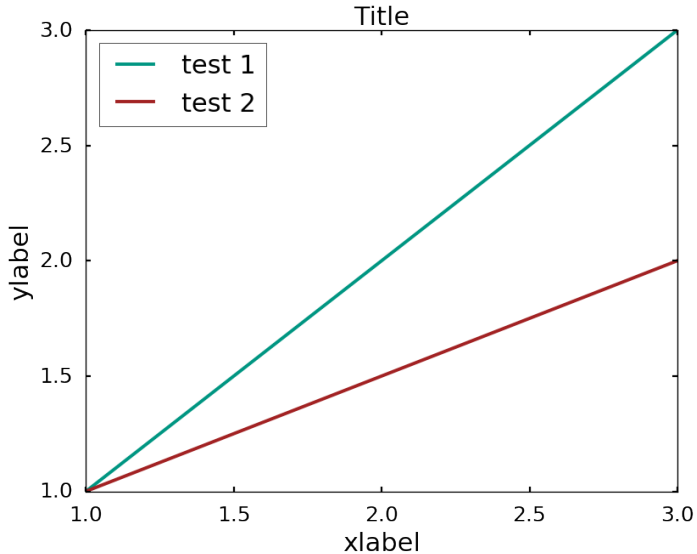
Colored style

Full style

```
from ekp_style import set_ekp_style
set_ekp_style(set_colors=True, set_background=True)

from matplotlib import pyplot as plt

plt.plot([1, 2, 3], [1, 2, 3], label="test 1")
plt.plot([1, 2, 3], [1, 1.5, 2], label="test 2")
plt.xlabel("xlabel")
plt.ylabel("ylabel")

plt.legend(loc=2, frameon=True)
plt.title("Title")
```

Full style