

# OdpDown Demo

---

[tbehrens@acm.org](mailto:tbehrens@acm.org)

# Advanced Markup

---

# Embedded Image Markup

---

## Embed images

```
![This is alt text]
(http://upload.wikimedia.org/wikipe
dia/commons/0/02/LibreOffice_Logo_F
lat.svg "This is an optional title
for a direct img")
```

# Embed images

---



This is an optional title for a direct img

# Referenced Images Markup

---

`## Embed images via reference`

`![This is alt text][2]`

`[2]:`

`https://wiki.documentfoundation.org  
/images/8/87/LibreOffice\_external\_l  
ogo\_600px.png "This is an optional  
title attribute for a ref img"`

# Embed images via reference

---



This is an optional title attribute for a ref img

# Inline Code Markup

---

## ## Inline Code

You can do html-alike inline `<code>` display with the following markup:

```
`$ tail -f /var/log/messages`
```

# Inline Code

---

You can do html-alike inline `<code>` display with the following markup: `$ tail -f /var/log/messages`

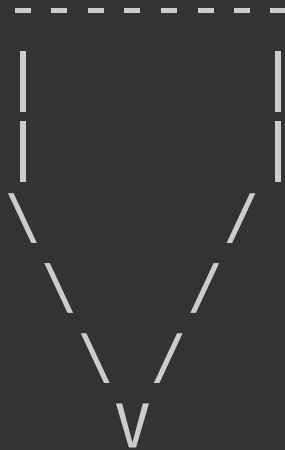


# Preformatted Content Markup

---

## ## Preformatted Content

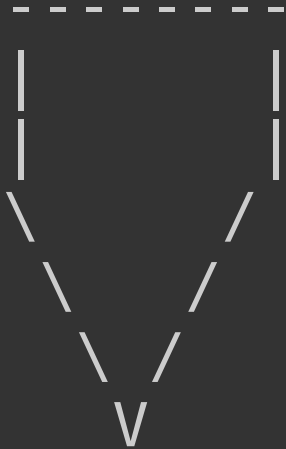
Just indent your text by four or more spaces, to have it rendered in monospaced as pre-formatted content:



# Preformatted Content

---

Just indent your text by four or more spaces,  
to have it rendered in monospaced as  
pre-formatted content:



# Syntax-Highlighted Code Markup

---

## Code - C++

By using a start/end marker as below (~~~ or ```), and specifying one of the 100+ supported pygments language identifiers (<http://pygments.org/languages/>):

```
~~~ c++
::basegfx::B2DPolyPolygon VeeWipe::operator () ( double t )
{
    ::basegfx::B2DPolygon poly;
    poly.append( ::basegfx::B2DPoint( 0.0, -1.0 ) );
    const double d = ::basegfx::pruneScaleValue( 2.0 * t );
    poly.append( ::basegfx::B2DPoint(
        0.0, d - 1.0 ) );
    poly.append( ::basegfx::B2DPoint(
        0.5, d ) );
    poly.append( ::basegfx::B2DPoint(
        1.0, d - 1.0 ) );
    poly.append( ::basegfx::B2DPoint(
        1.0, -1.0 ) );
    poly.setClosed(true);
    return ::basegfx::B2DPolyPolygon( poly );
}
~~~
```

# Code - C++

---

```
::basegfx::B2DPolyPolygon VeeWipe::operator () ( double t )
{
    ::basegfx::B2DPolygon poly;
    poly.append( ::basegfx::B2DPoint( 0.0, -1.0 ) );
    const double d = ::basegfx::pruneScaleValue( 2.0 * t );
    poly.append( ::basegfx::B2DPoint(
        0.0, d - 1.0 ) );
    poly.append( ::basegfx::B2DPoint(
        0.5, d ) );
    poly.append( ::basegfx::B2DPoint(
        1.0, d - 1.0 ) );
    poly.append( ::basegfx::B2DPoint(
        1.0, -1.0 ) );
    poly.setClosed(true);
    return ::basegfx::B2DPolyPolygon( poly );
}
```

# Or the same for syntax-highlighting Python

---

```
## Code - Python
```

```
~~~ python
# helper for ODFFormatter and ODFRenderer
def add_style(document, style_family, style_name,
              properties, parent=None):
    """Insert global style into given document"""
    style = odf_create_style(style_family,
                             style_name,
                             style_name,
                             parent)

    for elem in properties:
        # pylint: disable=maybe-no-member
        style.set_properties(properties=elem[1],
                             area=elem[0])
    document.insert_style(style, automatic=True)
~~~
```

# Code - Python

---

```
# helper for ODFFormatter and ODFRenderer
def add_style(document, style_family, style_name,
               properties, parent=None):
    """Insert global style into given document"""
    style = odf_create_style(style_family,
                             style_name,
                             style_name,
                             parent)

    for elem in properties:
        # pylint: disable=maybe-no-member
        style.set_properties(properties=elem[1],
                             area=elem[0])
    document.insert_style(style, automatic=True)
```

# Or for Bash-script syntax-highlighting

---

```
## Code - Bash
```

```
~~~ bash
# sanity checks
which safecat > /dev/null 2>&1 || {
    echo "You need safecat for this!"
    exit 1
}

umask 077

# enqueue mail, and params.
QUEUE_NAME=`safecat $BASE_DIR/tmp $BASE_DIR/mails`
if [ $? -eq 0 ]; then
    echo -e "$QUEUE_NAME\n$@" | \
        safecat $BASE_DIR/tmp $BASE_DIR/queue 1>/dev/null && \
        exit 0
    rm $BASE_DIR/mails/$QUEUE_NAME
fi

exit 1
~~~
```

# Code - Bash

---

```
# sanity checks
```

```
which safecat > /dev/null 2>&1 || {  
    echo "You need safecat for this!"  
    exit 1  
}
```

```
umask 077
```

```
# enqueue mail, and params.
```

```
QUEUE_NAME=`safecat $BASE_DIR/tmp $BASE_DIR/mails`  
if [ $? -eq 0 ]; then  
    echo -e "$QUEUE_NAME\n$@" | \  
        safecat $BASE_DIR/tmp $BASE_DIR/queue 1>/dev/null && \  
        exit 0  
    rm $BASE_DIR/mails/$QUEUE_NAME  
fi  
  
exit 1
```