

Jolis mails HTML automatiques

(ou les 7 cercles de l'enfer)

(ou “crash course pour les malheureux”)

Moi, très rapidement

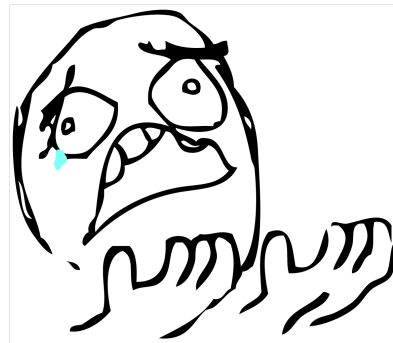
- Victor Perron
- vient tout juste d'arriver chez **Polyconseil**
- les deux dernières années au **Japon**
- me suis frotté aux joies de l'**i18n** & **l10n**
- sans oublier les **timezones** ...

Sachez apprécier la beauté des slides à venir.

Facile ? いいえ。

- parce qu'il y a des **images**.
- et mêmes des images **dynamiques**.
- parce qu'il faut prévoir la **version texte**.
- parce que l'on veut **plusieurs langues**.
- parce que le **CSS** dans les **clients mail**.

... et aussi parce que les **mobiles**.



Tips pour les images

- Pour tes amis sous **Retina**, prévois des images double taille puis réduis.

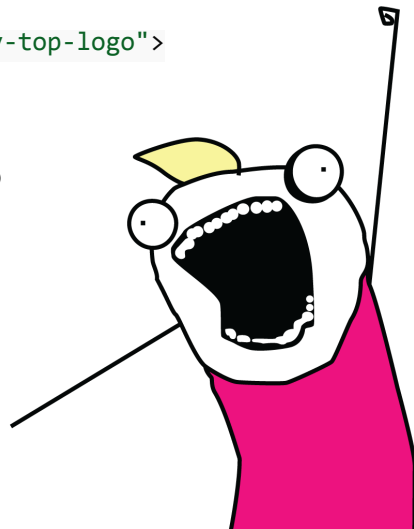
```

```

- **Troll**: src="<http://whatever.cdn/img>" or src="cid:img" ?

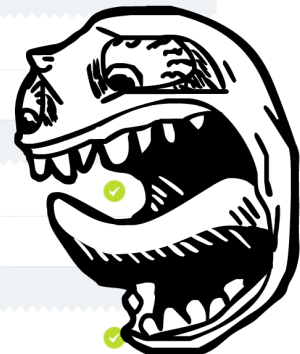
- N'hésite pas à utiliser **tous les headers** à la fois.

```
def attach_inline_image_data(msg, name, data, ext='png'):  
    msg_img = MIMEImage(data)  
    msg_img.add_header('Content-ID', '<{name}>'.format(name=name))  
    msg_img.add_header('X-Attachment-Id', '{name}'.format(name=name))  
    msg_img.add_header('Content-Disposition', 'inline; filename="{name}.{ext}"'.format(name=name, ext=ext))  
    msg_img.replace_header('Content-Type', 'image/{ext}; name="{name}.{ext}"'.format(name=name, ext=ext))  
    msg.attach(msg_img)
```



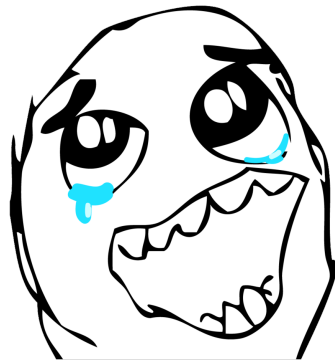
CSS FTW

Style Element	Outlook 2007/10/13 +	Outlook 03/Express/Mail	iPhone iOS 7/iPad	Outlook.com	Apple Mail 6.5	Yahoo! Mail	Google Gmail	Android 4 (Gmail) +
Responsive								
Responsive	x	x	✓	x	✓	x	x	x
Style Element								
<style> in <head>	✓	✓	✓	✓	✓	✓	x	
<style> in <body>	✓	✓	✓	✓	✓	✓	x	
Link Element								
<link> in <head>	✓	✓	✓	x	✓	x	x	✓
<link> in <body>	✓	✓	✓	x	✓	x	x	
Selectors								
*	x	✓	✓	x	✓	x	i	✓



- Le support du CSS dans les clients mail: **inégal** :’((
- Outlook, mobile, grid-based, ils l’ont fait pour nous: utiliser **Zurb/Ink**
- Presque aucun client ne reconnaît les classes CSS: utiliser **premailer**

Now With Django



- Templates **Ink** + **blocktrans** + **Jinja2**: on respire

```
<h6>{% trans "Traffic Conversion" %}</h6>
<p class="center">{% blocktrans %}
Your have done <b>{{transactions}}</b> !!!
{% endblocktrans %}
</p>
```

- *Follow the red highlights*

```
django.utils.translation.activate(user.locale)                # activate automagic translation (coz no requests)
html_message = render_to_string('email/weekly.html', context)
inlined_html = premailer.Premailer(html_message).transform()  # strip ALL the classes !
msg = EmailMultiAlternatives(subject, text_message, [user.email])
msg.attach_alternative(inlined_html, 'text/html')
msg.mixed_subtype = 'related'                                  # the ultimate tip: get embedded images...
driver = selenium.webdriver.PhantomJS()
driver.set_window_size(800, 600)                               # Let's take a fully server-generated screenshot :)
attach_inline_image_data(msg, 'my_image', driver.get_screenshot(context))
```

