Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Text Visualization with Wordcloud

Name: Phua Yan Han

IC Number: 050824070059

Date : 4/7/23

Introduction : learning word cloud

Conclusion : learned the way to shape wordcloud

P17 - Visualizing Text with Word Cloud

Word Cloud

What is a word cloud?

Data visualizations (like charts, graphs, infographics, and more) one of the many ways to communicate important information at a glance, but what if the raw data is text-based?

Word clouds (also known as text clouds or tag clouds): the more a specific word appears in a source of textual data (such as a speech, blog post, or database), the bigger and bolder it appears in the word cloud.

A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it’s mentioned within a given text and the more important it is.

Also known as tag clouds or text clouds, these are ideal ways to pull out the most pertinent parts of textual data, from blog posts to databases. They can also help business users compare and contrast two different pieces of text to find the wording similarities between the two.

Useful for quick summary of common customer feedback, text documents, identifying new SEO terms to target.

https://pypi.org/project/wordcloud/ (https://pypi.org/project/wordcloud/)

Know how to search for packages?

https://en.wikipedia.org/wiki/Tag_cloud (https://en.wikipedia.org/wiki/Tag_cloud)

References:

https://amueller.github.io/word_cloud/ (https://amueller.github.io/word_cloud/)

https://github.com/amueller/word_cloud (https://github.com/amueller/word_cloud)

https://www.kaggle.com/agisga/word-clouds (https://www.kaggle.com/agisga/word-clouds)

Installation

conda install -c conda-forge wordcloud

In [3]:
```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

text = "This is my first Word Cloud, Word Cloud is cool. Whatever this is"

wc = WordCloud()
wc = WordCloud(background_color="white", repeat=True)

wc.generate(text)

plt.axis("off")
plt.imshow(wc, interpolation="bilinear")
plt.show()
```

In [2]:
```
from wordcloud import WordCloud, STOPWORDS

STOPWORDS
```

Out[2]:
```
{'a',
 'about',
 'above',
 'after',
 'again',
 'against',
 'all',
 'also',
 'am',
 'an',
 'and',
 'any',
 'are',
 "aren't",
 'as',
 'at',
 'be',
 'because',
 'been',
 'before',
 'being',
 'below',
 'between',
 'both',
 'but',
 'by',
 'can',
 "can't",
 'cannot',
 'com',
 'could',
 "couldn't",
 'did',
 "didn't",
 'do',
 'does',
 "doesn't",
 'doing',
 "don't",
 'down',
 'during',
 'each',
 'else',
 'ever',
 'few',
 'for',
 'from',
 'further',
 'get',
 'had',
 "hadn't",
 'has',
 "hasn't",
 'have',
 "haven't",
 'having',
 'he',
```

```
                "he'd",
                "he'll",
                "he's",
                'hence',
                'her',
                'here',
                "here's",
                'hers',
                'herself',
                'him',
                'himself',
                'his',
                'how',
                "how's",
                'however',
                'http',
                'i',
                "i'd",
                "i'll",
                "i'm",
                "i've",
                'if',
                'in',
                'into',
                'is',
                "isn't",
                'it',
                "it's",
                'its',
                'itself',
                'just',
                'k',
                "let's",
                'like',
                'me',
                'more',
                'most',
                "mustn't",
                'my',
                'myself',
                'no',
                'nor',
                'not',
                'of',
                'off',
                'on',
                'once',
                'only',
                'or',
                'other',
                'otherwise',
                'ought',
                'our',
                'ours',
                'ourselves',
                'out',
                'over',
```

```
                'own',
                'r',
                'same',
                'shall',
                "shan't",
                'she',
                "she'd",
                "she'll",
                "she's",
                'should',
                "shouldn't",
                'since',
                'so',
                'some',
                'such',
                'than',
                'that',
                "that's",
                'the',
                'their',
                'theirs',
                'them',
                'themselves',
                'then',
                'there',
                "there's",
                'therefore',
                'these',
                'they',
                "they'd",
                "they'll",
                "they're",
                "they've",
                'this',
                'those',
                'through',
                'to',
                'too',
                'under',
                'until',
                'up',
                'very',
                'was',
                "wasn't",
                'we',
                "we'd",
                "we'll",
                "we're",
                "we've",
                'were',
                "weren't",
                'what',
                "what's",
                'when',
                "when's",
                'where',
                "where's",
```

```
 'which',
 'while',
 'who',
 "who's",
 'whom',
 'why',
 "why's",
 'with',
 "won't",
 'would',
 "wouldn't",
 'www',
 'you',
 "you'd",
 "you'll",
 "you're",
 "you've",
 'your',
 'yours',
 'yourself',
 'yourselves'}
```

Let's get real world data

From Wikipedia

conda install -c conda-forge wikipedia

In [1]:
```
import sys
import wikipedia
from wordcloud import WordCloud, STOPWORDS

inputstring = str(input('Enter the title; '))

title = wikipedia.search(inputstring)[0]

page = wikipedia.page(title)

text = page.content
```

```
Enter the title; Tears of the Kingdom
```

In [2]:
```
print(text)
```

The Legend of Zelda: Tears of the Kingdom is a 2023 action-adventure game dev eloped and published by Nintendo for the Nintendo Switch. The sequel to The L egend of Zelda: Breath of the Wild (2017), Tears of the Kingdom retains aspec ts including the open world of Hyrule, which has been expanded to allow for m ore vertical exploration. The player controls Link as he searches for Princes s Zelda and fights to prevent the Demon King from destroying the world.
Tears of the Kingdom was conceived after ideas for Breath of the Wild downloa dable content (DLC) had exceeded its scope. Its development was led by Ninten do's Entertainment Planning & Development (EPD) division, with Breath of the Wild director Hidemaro Fujibayashi and producer Eiji Aonuma reprising their r oles. A teaser was shown at E3 2019 with a full reveal at E3 2021. Tears of t he Kingdom was initially planned for release in 2022 before being delayed to May 2023. It received acclaim for its improvements, expanded open world, and features encouraging exploration and experimentation. It sold more than 10 mi llion copies in its first three days of release.

== Gameplay ==
Tears of the Kingdom retains the open-world action-adventure gameplay of the previous Zelda game, Breath of the Wild (2017). As Link, players explore Hyru le and two new areas, the sky, littered with numerous floating islands, and t he Depths, to find weapons, resources, and complete quests. Link can explore on foot or by climbing, horseriding and using paragliders.New to Tears of the Kingdom are the Zonai devices, which the player can use for combat, propulsio n, exploration, and more. The previous game's runes are replaced with five ne w powers: Ultrahand, Fuse, Ascend, Recall, and Autobuild. Ultrahand allows th e player to pick up and move different objects, and attach different objects together. This can be used with the Zonai devices to create different vehicle s or other constructs. Fuse allows Link to combine materials, equipment, or c ertain objects in the world to a shield or a weapon to increase its attribute s and durability. For example, fusing an explosive object to an arrow will ca use the arrow to explode on impact. Autobuild instantly recreates a device cr afted with Ultrahand, automatically using nearby devices and objects if avail able, or if parts are missing, creating single use replacements at the cost o f zonaite. Ascend allows the player to move upwards through solid surfaces. R ecall can be used on an object to "rewind" its movement.Shrines and Korok see ds return. When cleared, shrines grant players a Light of Blessing; when four are obtained, Link can use them to either increase the amount of hearts he ha s by one or expand the size of his stamina wheel. Korok seeds can be used to increase the inventory capacity of Link, allowing him to hold more melee weap ons, shields, or bows.
The Depths, which are accessed via chasms on the surface, house a new threat, a goopy substance called gloom. Whenever Link comes into contact with or is a ttacked by enemies covered in gloom, his maximum number of hearts temporarily decreases by one. This affliction is removed only when Link returns to the su rface or travels to a lightroot, a large light-emitting plant found in the De pths, to regain his maximum health.

== Plot ==

Tears of the Kingdom takes place a number of years after Breath of the Wild, at the end of the Zelda timeline in the kingdom of Hyrule. Link and Zelda set out to explore a cavern beneath Hyrule Castle, from which gloom, a poisonous substance, has been seeping out and causing people to fall ill. There, they f ind a mural depicting a conflict known as the Imprisoning War—an ancient batt le against a being only referred to as the "Demon King"—and decide to venture

deeper, triggering the awakening of a mummy who attacks the two with gloom. L
ink's right arm is badly damaged, the Master Sword is shattered, and Hyrule C
astle is raised upwards by the mummy. Zelda falls below, and as Link tries to
catch her, she vanishes with a mysterious artifact. Link is rescued by a dise
mbodied arm, which had been restraining the mummy, and on the Great Sky Islan
d, he awakens to find that it has replaced his damaged arm. He meets the spir
it of Rauru, a Zonai, and the source of Link's new arm. Rauru assists Link as
he traverses the Great Sky Island, and when the latter reaches his destinatio
n, the shattered Master Sword vanishes, and Link dives to the surface below.
There, he learns that the event in the cavern, known as the Upheaval, has wro
ught chaos upon Hyrule, and sets out to investigate reports of strange phenom
ena throughout the kingdom. In the process, he reunites with Sidon, the princ
e—later, king—of the Zora; Tulin, a young Rito archer; Yunobo, a Goron and th
e president of YunoboCo, a mining company; and Riju, the chief of the Gerudo,
and alongside them, Link ventures into ancient Zonai temples. With the defeat
of the monsters occupying the temples, Link finds strange artifacts. The spir
its of sages from ancient times appear to Link and his allies, appointing the
latter as sages and passing down the artifacts, secret stones, which they onc
e wielded. Afterwards, after reports of Zelda's appearance across Hyrule, Lin
k tracks her down to Hyrule Castle, where she reveals herself to be an impost
or serving the Demon King, Ganondorf, who was the mummy beneath Hyrule Castl
e. The impostor, Phantom Ganon, is defeated by Link and his allies, and the f
ormer later encounters Mineru, another ancient sage, who remains in the physi
cal world via spiritual projection. He helps create a mechanical body for her
spirit to inhabit.
Through Mineru, the other ancient sages, and Dragon's Tears scattered through
out the land, Link learns of Zelda's fate. The artifact that vanished with Ze
lda was a secret stone, and with it, she was transported through time to the
distant past. There, she meets Rauru, revealed to be Hyrule's first king, and
Sonia, Hyrule's first queen. In the past, Ganondorf kills Sonia, stealing her
secret stone and becoming the Demon King. Rauru appoints Mineru, Zelda, and t
he leaders of the Zora, Rito, Gorons, and Gerudo as sages, and sacrifices him
self to seal Ganondorf. Later, Zelda receives the Master Sword and tasks the
sages to aid Link in the future. In order to restore the Master Sword and bri
ng it to Link, she swallows her secret stone and undergoes a process known as
draconification, becoming the Light Dragon.
In the present, after ridding the Great Deku Tree from gloom in Korok Forest,
Link retrieves the Master Sword from the Light Dragon and goes to confront Ga
nondorf below Hyrule Castle. With the aid of Sidon, Tulin, Yunobo, Riju, and
Mineru, Link battles an army of monsters before engaging in combat with Ganon
dorf himself. Nearing defeat, Ganondorf swallows his secret stone and becomes
the Demon Dragon, taking Link into the skies above Hyrule. With the assistanc
e of the Light Dragon, Link uses the Master Sword to shatter the Demon Drago
n's secret stone, killing it. Aided by the spirits of Rauru and Sonia, Link t
urns Zelda back to normal and regains his own right arm. Rauru and Sonia fade
away, and Link and Zelda fall to the surface below, where they reunite.
Some time later, on the Great Sky Island, Mineru bids goodbye to Zelda and Li
nk before fading away, while the sages vow to protect Hyrule.

== Development ==

Development of Tears of the Kingdom started in 2017 after Breath of the Wild
was completed. Initially, new ideas were thought of for DLC but eventually it
evolved into a new game when too many were gathered. It was announced at E3 2
019 as a sequel to Breath of the Wild. At E3 2021, Nintendo debuted a trailer
revealing gameplay, story elements and a 2022 release window, but Nintendo la

ter changed the release window to Q2 2023. More information was revealed in t
he Nintendo Direct presentation held in September 2022, including the title T
ears of the Kingdom and a release date of May 12, 2023. Another Nintendo Dire
ct in February 2023 teased more gameplay elements. A playable version of the
game leaked online as a disk image two weeks before release.As with Breath of
the Wild, Tears of the Kingdom was directed by Hidemaro Fujibayashi and produ
ced by Eiji Aonuma. It was conceived after the team was unable to use every i
dea planned for Breath of the Wild's downloadable content. New elements inclu
de floating islands above Hyrule, with players able to soar between them in a
style similar to The Legend of Zelda: Skyward Sword (2011). While he was show
n early demos, the involvement of the Zelda creator, Shigeru Miyamoto, was re
duced due to scheduling conflicts with his role as a producer on The Super Ma
rio Bros. Movie (2023). He was credited as "general producer", the same posit
ion he had in the series since the end of the 2000s.The development team crea
ted ideas during the development of Breath the Wild. Aonuma said that the tea
m agreed that the sequel would return to the same version of Hyrule. The tech
nical director, Takuhiro Dohta, said that he was inspired by Wii Sports Resor
t to use the same place but add new mechanics: "The idea of having new discov
eries in the same setting was striking to me." He also noted that using famil
iar locations is useful for players when diving from the sky. The team extend
ed the exploration areas into the sky and underground. Aonuma referred to Sky
ward Sword commenting that due to the limitations of the hardware the develop
ers were not able to achieve a seamless descent from the sky to the surface.
In that game, Link was limited to diving only from specific points. With the
capabilities of the Switch, Tears of the Kingdom could now provide the freedo
m of traversing a game world that is connected both horizontally and vertical
ly.Due to its similarity to Breath of the Wild, the developers experienced "s
trong deja vu". Aonuma said that the development team was trying to create so
mething new but also something similar to the previous game and realised that
some aspects "were already as they should be". Fujibayashi also said that the
re were occasions where they struggled to differentiate between the two game
s. The developers approached Tears of the Kingdom with a list of ideas that t
hey had wanted to include in Breath of the Wild. Aonuma wanted the player to
go underground in Breath of the Wild but was restricted by the technical limi
tations of the Wii U. Tears of the Kingdom reintroduced large dungeons into t
he series, which are connected to Hyrule's surface. Technical director Takuhi
ro Dohta said that the dungeons were designed with regional characteristics t
o make them unique to their respective environments "just like traditional Th
e Legend of Zelda games". The dungeons were primarily created to showcase the
range of Link's powers and gadgets as a way to maximize the gameplay. Additio
nally, they were designed to be accessed seamlessly rather than being closed
off, allowing the player to dive from the sky straight down into the dungeon
and step out again at any point.One of the core concepts is the ability to bu
ild new items. Fujibayashi said that immediately after the release of Breath
of the Wild players began experimenting and posted videos of their accomplish
ments on social media. This gave the development team the confidence to creat
e more tools in Tears of the Kingdom that players could use to create their o
wn unique gameplay experience. Zonai devices were introduced to encourage gre
ater experimentation, but this also required a balance to be set between allo
wing the player to be creative and maintaining limitations to ensure players
were not able to break gameplay with infinitely powered devices.The theme of
"hands" was used as an overarching element as a way to express the concept of
connecting. It is present in the mechanics, story, visuals and sound. Dohta s
aid that "joining hands" and cooperating with other characters is a major asp
ect, along with Link's ability to use his hands to create items. Link's right
arm was created as a way to distinguish him from previous iterations of the c
haracter, but is also used symbolically to solve puzzles and open doors. Aonu

ma said the theme is also present within the story, which involves connecting to Hyrule's history.The Ascend ability was developed from a debug feature tha t Aonuma and Fujibayashi used to exit the caves. They agreed that it was tire some to navigate all the way through the caves and wanted a cheat to get to t he surface. Its implementation posed other challenges, such as ensuring that the player would not ascend into an empty space due to data loading problems. Aonuma cited inspiration from the open-world games Red Dead Redemption 2 and The Elder Scrolls V: Skyrim. By March 2022, Tears of the Kingdom was essentia lly complete, but Nintendo delayed it for a year to refine it.

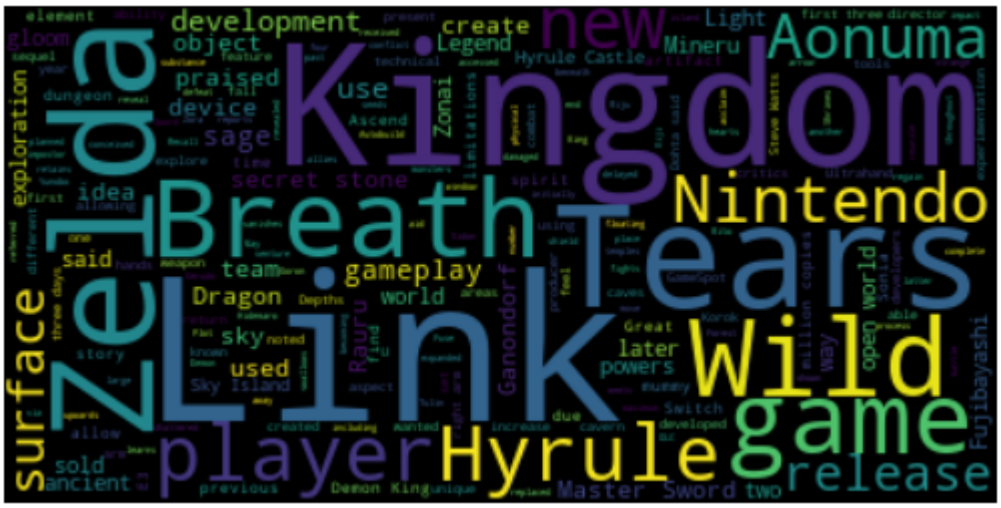
== Reception ==

According to the review aggregator Metacritic, Tears of the Kingdom received "universal acclaim".Several critics praised the expansion of the open world i ntroduced in Breath of the Wild. Tom Marks of IGN praised the addition of the sky islands and caves as "massive" and "brilliant complements to the more tra ditional surface activities". He also complimented the story as distinguishin g itself from typical plots of Zelda games. Edwin Evans-Thirwell of Eurogamer praised the seamlessness of the "loop between underworld and sky", writing th at while the "caverns aren't always worth the toil of discovery, none of the sky islands feel dispensable", praising the "irresistible, toylike specificit y" of their design. On the differences from Breath of the Wild, Steve Watts o f GameSpot praised the "subtle ways" the world had changed, saying that "not everything is exactly the same or where you'd expect it to be, and the map is marked with myriad opportunities for exploration and curiosity". Game Informe r's Kyle Hilliard "didn't get the same goosebumps exploring Hyrule" as he did from Breath of the Wild, but nevertheless "adored returning to Hyrule with al l new tools."Critics praised the addition of powers and building features to traverse and interact with the environment. Tom Marks of IGN wrote that "the sandbox is bigger, richer, and somehow even more ambitious", praising the bui lding system as "walking the line of powerful but approachable extremely wel l", "tons of fun", and "woven into every part". Steve Watts of GameSpot descr ibed the powers as "much more flexible" than its predecessor and a "beautiful ly implemented evolution of what made Breath of the Wild so special", stating "these tools give Tears of the Kingdom a particular flow that feels unique to the Zelda franchise." Mike Mahardy of Polygon praised the emphasis on "self-d riven experimentation" as "paramount in traversal, combat, and discovery ... Experimenting with each of [the] powers and discovering how they interact is the game."Several critics were mixed on the impact of performance problems. A lana Hagues of Nintendo Life wrote: "It's evident that Tears of the Kingdom i s pushing the system to its limits", citing frame rate drops to the "low 20s" during "busy fights and locations", although stating "it's not hugely disrupt ive, and it didn't feel any worse than Breath of the Wild ... but it serves a s a staunch reminder of the now-six-year-old console's limitations." Writing that "the Switch continues to show its age", Mike Mahardy of Polygon noted "t he frame rate can still be abysmal, especially in dense forests or areas with busy water effects. Loading times are unpredictable, ranging from two seconds to 10." Steve Watts of GameSpot noted concerns about the aging Switch hardwar e, saying he "barely ever saw a hitch while playing mostly in handheld mode ... even then, the performance dips were minor and temporary."

=== Sales ===
Tears of the Kingdom was the first Nintendo-developed game to be priced at $7 0 in the US. More than 10 million copies of Tears of the Kingdom were sold in its first three days of release, making it the fastest-selling game in The Le

```
gend of Zelda franchise, as well as the fastest selling Nintendo game in the
Americas with over four million copies sold in the US alone. Tears of the Kin
gdom sold over 2.24 million copies within its first three days of release in
Japan, and according to Famitsu, it sold over 1.1 million physical copies alo
ne.


=== Accolades ===


== Notes ==


== References ==
```

In [3]:
```
import matplotlib.pyplot as plt

wordcloud  = WordCloud(background_color='black', max_words=200, stopwords=STOPW

wordcloud.generate(text)

plt.imshow(wordcloud,interpolation='bilinear')
plt.axis("off")
plt.show()
```



2. From PDF File

In [4]:
```
import requests

url = 'https://www.agc.gov.my/agcportal/common//uploads/publication/391/2020_11

# Download the PDF
myfile = requests.get(url, allow_redirects=True, verify=False)

open('./IT_Security_Policy_for_AGC.pdf', 'wb').write(myfile.content)
```

```
C:\Users\Asus\anaconda3\envs\python-dscourse\lib\site-packages\urllib3\connec
tionpool.py:1056: InsecureRequestWarning: Unverified HTTPS request is being m
ade to host 'www.agc.gov.my'. Adding certificate verification is strongly adv
ised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-w
arnings (https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-war
nings)
  warnings.warn(
```

Out[4]: 1485266

In [1]:
```
# Convert PDF to Text
import PyPDF2

with open('IT_Security_Policy_for_AGC.pdf','rb') as pdf_file, open('IT_Security
    read_pdf = PyPDF2.PdfFileReader(pdf_file)
    number_of_pages = read_pdf.getNumPages()
    for page_number in range(number_of_pages):
        page = read_pdf.getPage(page_number)
        page_content = page.extractText()
        text_file.write(page_content)
```

In [2]:
```
page_content
```

Out[2]:
```
'DASAR KESELAMATAN TE KNOLOGI MAKLUMAT JAB ATAN PEGUAM NEGARA   \nTARIKH : 15
FEBRUARI  2018 MUKA SURAT  74 DARI 75 \n o) Akta Rahsia Rasmi 1972;  \n \np)
Akta Jenayah Komputer 1997;  \n \nq) Akta Hak Cipta (Pindaan) Tahun 1997;  \n
\nr) Akta Komunikasi dan Multimedia 1998;  \n \ns) Perintah -Perintah Am;  \n
\nt)  Arahan Perbendaharaan;  \n \nu) Arahan Teknologi Maklumat 2007;  \n \n
v)  Garis Panduan Keselamatan AGC 2004;  \n \nw) Standard Operating Procedure
(SOP) ICT AGC;  \n \nx) Surat Pekeliling Am Bilangan 3 Tahun 2009 – Garis Pan
duan Penilaian Tahap \nKeselamatan Rangkaian dan Sistem ICT Sektor Awam yang
bertarikh 17 \nNovember 2009;  \n \ny) Surat Arahan Peguam Negara AGC – Pengu
rusan Kesinambungan \nPerkhidmatan Ag ensi Sektor Awam yang bertarikh 22 Janu
ari 2010.  \n '
```

Alternative PDF libraries

https://anaconda.org/anaconda/repo (https://anaconda.org/anaconda/repo)

http://mstamy2.github.io/PyPDF2/ (http://mstamy2.github.io/PyPDF2/)

https://pypi.org/project/pdftotext/ (https://pypi.org/project/pdftotext/)

https://realpython.com/pdf-python/ (https://realpython.com/pdf-python/)

Downloading Files

https://dzone.com/articles/simple-examples-of-downloading-files-using-python

In [ ]:
```
# %matplotlib inline
```

In [3]:
```
from wordcloud import WordCloud, STOPWORDS

# Read the whole text.
text = open('./IT_Security_Policy_for_AGC.txt', encoding='utf-8').read()
text = text.replace ('  ',' ')

# Generate a word cloud image
wordcloud = WordCloud().generate(text)

# Display the generated image:

# the matplotlib way:
import matplotlib.pyplot as plt
plt.axis("off")
plt.imshow(wordcloud, interpolation='bilinear')
plt.show()

# The pil way (if you don't have matplotlib)
# from IPython.display import Image
# pil_img = wordcloud.to_image()
# display(pil_img)
```

In [4]:
```
# Generate a word cloud image
wordcloud = WordCloud().generate(text)

# Display the generated image:
plt.figure(figsize=(10,10)) #inches
plt.axis("off")
plt.imshow(wordcloud, interpolation='bilinear')

plt.show()

# note image size generated and the canvas size of plot
# https://matplotlib.org/3.2.1/api/_as_gen/matplotlib.pyplot.figure.html
```

In [5]:
```
# Generate a word cloud image
wordcloud = WordCloud(width=800, height=400).generate(text)
#wordcloud = WordCloud(width=3600, height=1600).generate(text)

# Display the generated image:
plt.figure(figsize=(10,10)) # inches
plt.axis("off")
plt.imshow(wordcloud, interpolation='bilinear')
plt.show()

# note image size generated and the canvas size of plot
```

In [6]:
```
# Lower max_font_size
wordcloud = WordCloud(max_font_size=20).generate(text)

# Display the generated image:
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

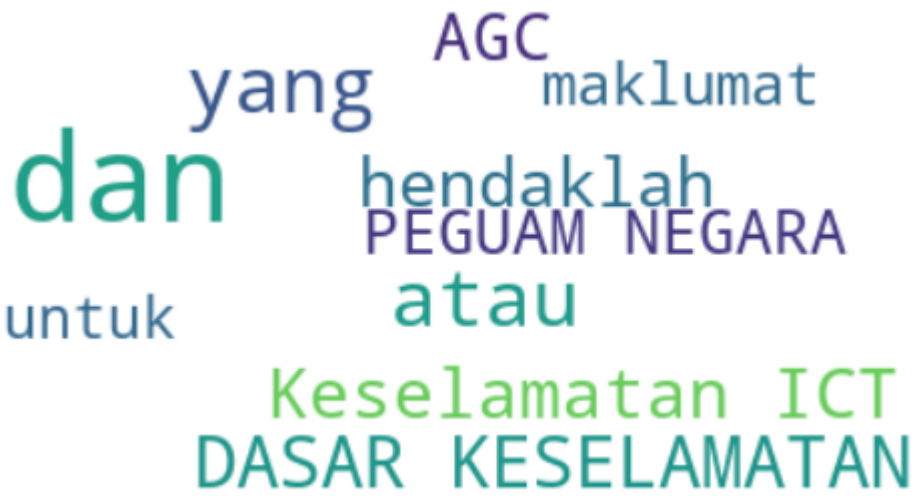
Out[6]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

In [7]:
```
# Change font size, Background Color
wordcloud = WordCloud(max_font_size=50, background_color='white').generate(text
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[7]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

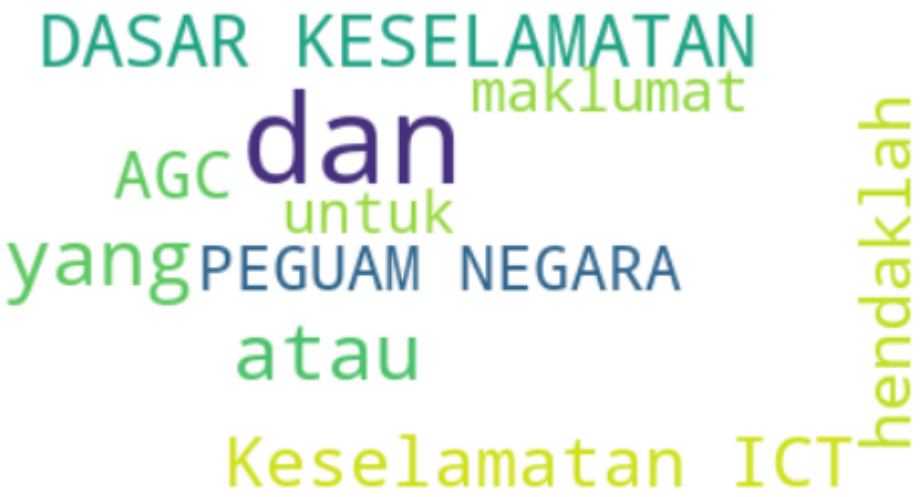
In [8]:
```
# Lower font size, maximum words, Background Color
wordcloud = WordCloud(max_font_size=50, max_words=10,background_color='white').
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[8]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



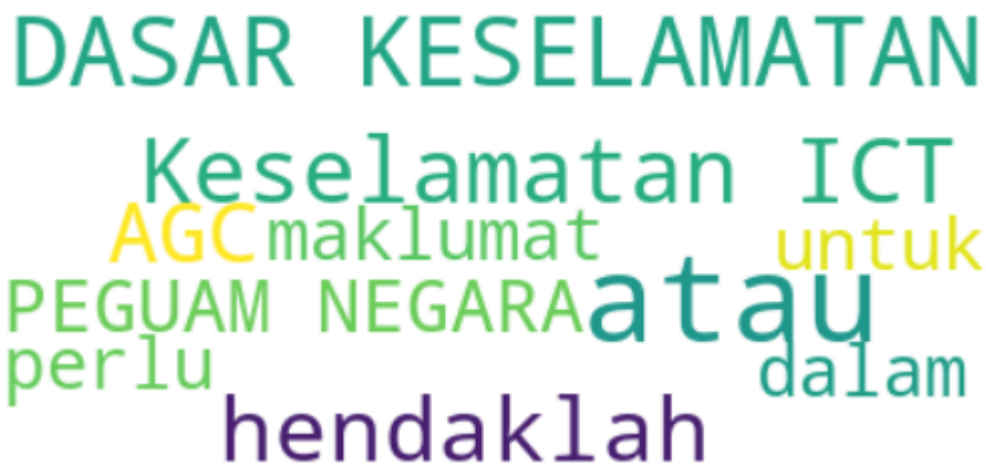
In [9]:
```
from wordcloud import STOPWORDS

# Create stopword list:
stopwords = set(STOPWORDS)

wordcloud = WordCloud(stopwords=stopwords, max_font_size=50, max_words=10,backg
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[9]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [11]:
```
from wordcloud import STOPWORDS

# Create stopword list:
stopwords = set(STOPWORDS)
stopwords.update(["dan", "yang","di","sabah","sarawak", "section","force","clau
# stop_words = list(stopwords)+["yang", "di","sabah sarawak", "section force",

wordcloud = WordCloud(stopwords=stopwords, max_font_size=50, max_words=10,backg
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[11]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [12]:
```
stopwords
```

Out[12]:
```
{'a',
 'about',
 'above',
 'act',
 'after',
 'again',
 'against',
 'agong',
 'all',
 'also',
 'am',
 'an',
 'and',
 'any',
 'are',
 "aren't",
 'as',
 'at',
 'be',
 'because',
 'been',
 'before',
 'being',
 'below',
 'between',
 'both',
 'but',
 'by',
 'can',
 "can't",
 'cannot',
 'clause',
 'com',
 'consitution',
 'could',
 "couldn't",
 'dan',
 'di',
 'did',
 "didn't",
 'do',
 'does',
 "doesn't",
 'doing',
 "don't",
 'down',
 'during',
 'each',
 'else',
 'ever',
 'federal',
 'few',
 'for',
 'force',
 'from',
 'further',
 'get',
```

```
'had',
"hadn't",
'has',
"hasn't",
'have',
"haven't",
'having',
'he',
"he'd",
"he'll",
"he's",
'hence',
'her',
'here',
"here's",
'hers',
'herself',
'him',
'himself',
'his',
'how',
"how's",
'however',
'http',
'i',
"i'd",
"i'll",
"i'm",
"i've",
'if',
'in',
'into',
'is',
"isn't",
'it',
"it's",
'its',
'itself',
'just',
'k',
"let's",
'like',
'me',
'more',
'most',
"mustn't",
'my',
'myself',
'no',
'nor',
'not',
'of',
'off',
'on',
'once',
'only',
'or',
```

```
                'other',
                'otherwise',
                'ought',
                'our',
                'ours',
                'ourselves',
                'out',
                'over',
                'own',
                'pertuan',
                'r',
                'sabah',
                'same',
                'sarawak',
                'section',
                'shall',
                "shan't",
                'she',
                "she'd",
                "she'll",
                "she's",
                'should',
                "shouldn't",
                'since',
                'so',
                'some',
                'such',
                'than',
                'that',
                "that's",
                'the',
                'their',
                'theirs',
                'them',
                'themselves',
                'then',
                'there',
                "there's",
                'therefore',
                'these',
                'they',
                "they'd",
                "they'll",
                "they're",
                "they've",
                'this',
                'those',
                'through',
                'to',
                'too',
                'under',
                'until',
                'up',
                'very',
                'was',
                "wasn't",
                'we',
```

```
"we'd",
"we'll",
"we're",
"we've",
'were',
"weren't",
'what',
"what's",
'when',
"when's",
'where',
"where's",
'which',
'while',
'who',
"who's",
'whom',
'why',
"why's",
'with',
"won't",
'would',
"wouldn't",
'www',
'yang',
'you',
"you'd",
"you'll",
"you're",
"you've",
'your',
'yours',
'yourself',
'yourselves'}
```

In [13]:
```
from wordcloud import WordCloud, STOPWORDS

testtext = 'yang di is'

# Create stopword list:
stopwords = STOPWORDS
stop_words = ['yang'] + list(stopwords)

wordcloud = WordCloud(stopwords=stop_words, max_font_size=50, max_words=10,back

plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[13]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```


Mask

Change the layout

Generate a Numpy grid

In [18]:
```
# Add Mask
import numpy as np

x, y = np.ogrid[:300, :300]

mask = (x - 150) ** 2 + (y - 150) ** 2 > 130 ** 2
mask = 255 * mask.astype(int)

wordcloud = WordCloud(max_font_size=50,background_color='white', mask=mask).ger
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
plt.savefig('mask.png')
```

Mask from another Image

First find or create an Image

Eg.

1. Use Paint and save it as mask.png

In [19]:
```
from IPython.display import display, Image
display(Image(filename='./mask.png'))
```

In [34]:
```
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

mask = np.array(Image.open('./yellow-house-to.png'))

color= ImageColorGenerator(mask)

wordcloud = WordCloud(width=50,
                      height=50,
                      max_words=100000,
                      mask=mask,
                      stopwords=STOPWORDS,
                      background_color='white',
                      random_state=42).generate(text)

plt.figure(figsize=(20,20)) # inches
plt.axis("off")
plt.imshow(wordcloud.recolor(color_func=color), interpolation='bilinear')
plt.show()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[34], line 6
      3 from PIL import Image
      4 import matplotlib.pyplot as plt
----> 6 mask = np.array(Image.open('./yellow-house-to.png'))
      8 color= ImageColorGenerator(mask)
     10 wordcloud = WordCloud(width=50,
     11                       height=50,
     12                       max_words=100000,
   (...)
     15                       background_color='white',
     16                       random_state=42).generate(text)

File ~\anaconda3\envs\python-dscourse\lib\site-packages\PIL\Image.py:3227,
in open(fp, mode, formats)
   3224     filename = fp
   3226 if filename:
-> 3227     fp = builtins.open(filename, "rb")
```

Read up

https://matplotlib.org/gallery/images_contours_and_fields/interpolation_methods.html (https://matplotlib.org/gallery/images_contours_and_fields/interpolation_methods.html)

2. Or download an Image

Flag of Malaysia

User Google Search

Find Images with larger sizes

Eg. https://en.wikipedia.org/wiki/Flag_of_Malaysia#/media/File:Flag_of_Malaysia.svg (https://en.wikipedia.org/wiki/Flag_of_Malaysia#/media/File:Flag_of_Malaysia.svg)

In [20]:
```
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

mask = np.array(Image.open('./1920px-Flag_of_Malaysia.svg.png'))

color= ImageColorGenerator(mask)

wordcloud = WordCloud(width=1920,
                      height=1080,
                      max_words=400,
                      mask=mask,
                      stopwords=STOPWORDS,
                      background_color='white',
                      random_state=42).generate(text)

plt.figure(figsize=(10,10)) # inches
plt.axis("off")
plt.imshow(wordcloud.recolor(color_func=color),interpolation='bilinear')
plt.show()
```

In [21]:
```
# Save to File
wordcloud.to_file('MalaysiaWordCloud.png')
```

Out[21]:
```
<wordcloud.wordcloud.WordCloud at 0x1b1e31fd190>
```

Try all the examples below

Python script to search google and produce a word cloud from the abstracts of the first page of results

https://github.com/charlie9578/googleWordCloud (https://github.com/charlie9578/googleWordCloud)

In [22]:
```
from wordcloud import WordCloud, STOPWORDS
from PIL import Image
import urllib
import requests
import numpy as np
import matplotlib.pyplot as plt

words = 'access guest guest apartment area area bathroom bed bed bed bed bed be
mask = np.array(Image.open(requests.get('http://www.clker.com/cliparts/O/i/x/Y/

# This function takes in your text and your mask and generates a wordcloud.
def generate_wordcloud(words, mask):
    word_cloud = WordCloud(width = 512, height = 512, background_color='white',
    plt.figure(figsize=(10,8),facecolor = 'white', edgecolor='blue')
    plt.imshow(word_cloud)
    plt.axis('off')
    plt.tight_layout(pad=0)
    plt.show()

#Run the following to generate your wordcloud
generate_wordcloud(words, mask)
```

Download from the source

The source code of word_cloud https://github.com/amueller/word_cloud (https://github.com/amueller/word_cloud)

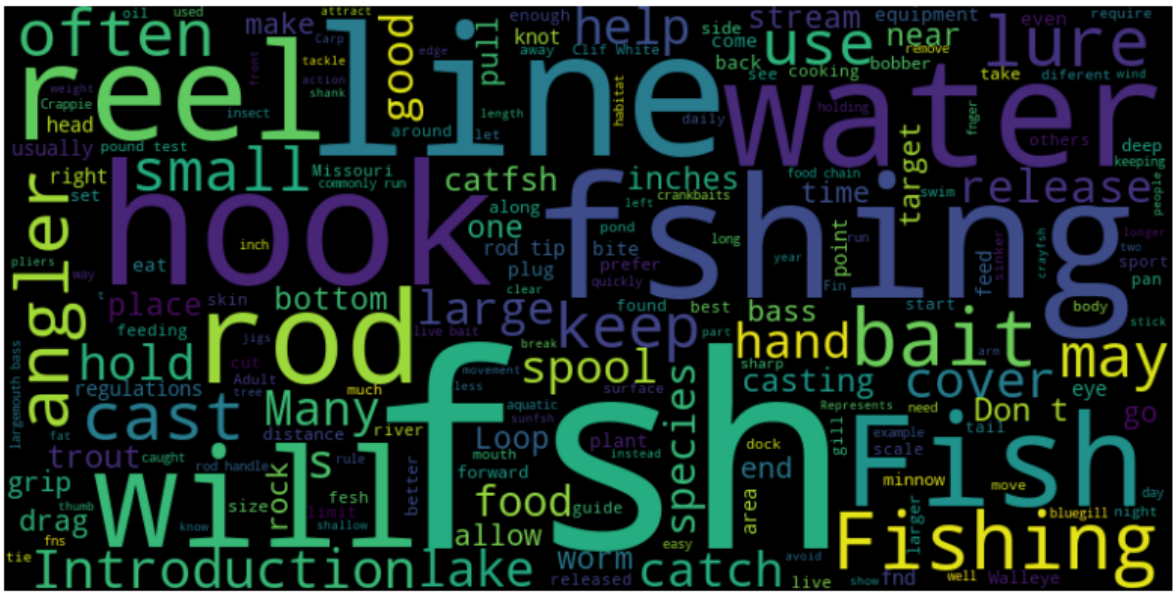
The Jupyter notebooks https://amueller.github.io/word_cloud/ (https://amueller.github.io/word_cloud/)

Quiz

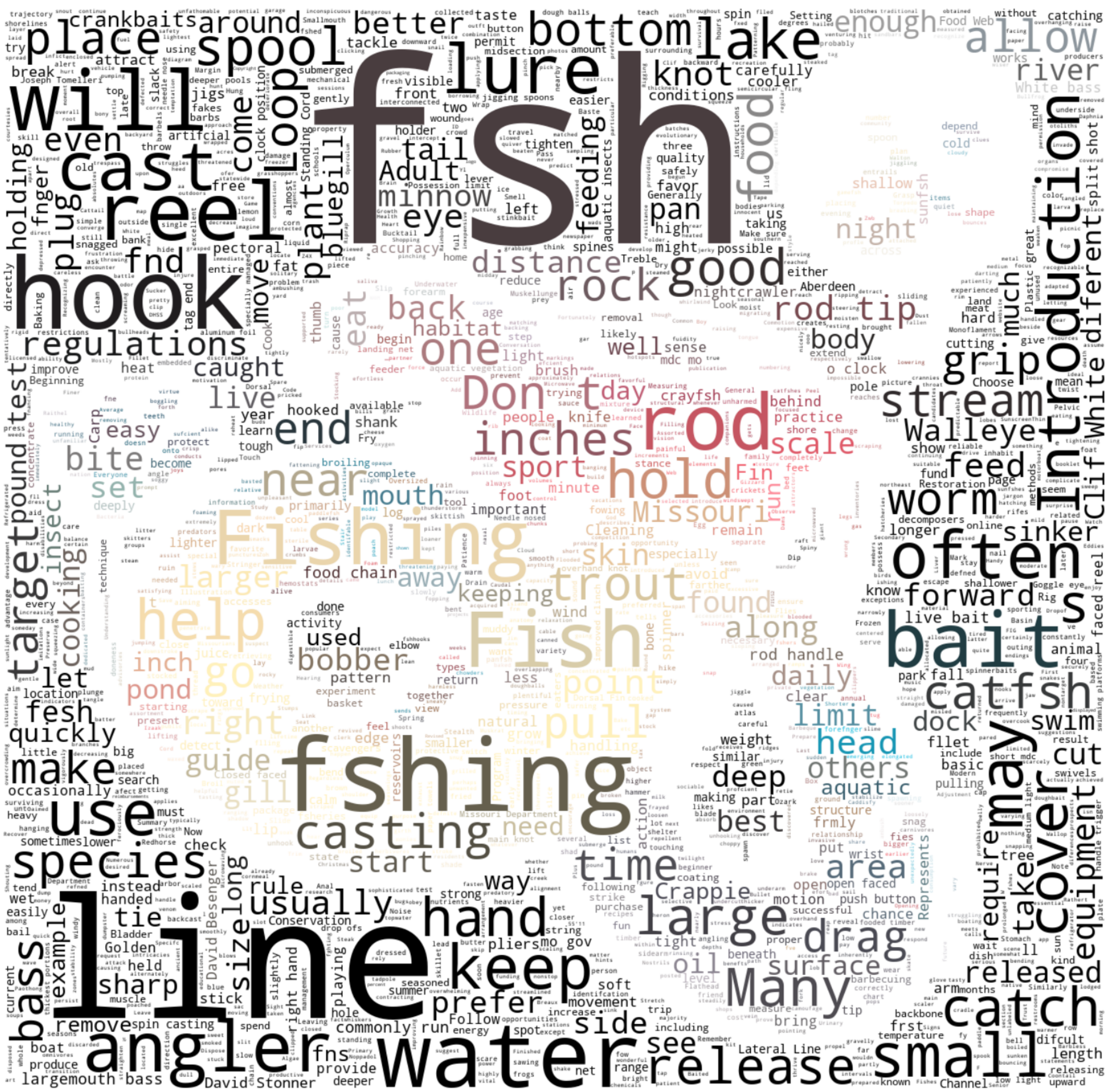
1. Download pdf from this link: https://huntfish.mdc.mo.gov/sites/default/files/downloads/page/IntroToFishing_2017_v2.pdf (https://huntfish.mdc.mo.gov/sites/default/files/downloads/page/IntroToFishing_2017_v2.pdf)
2. Text Visualization without mask for this text (using WordCloud)(Black and White)
3. Text Visualization with a mask (you can choose your prefered mask)

- Put in the url link of your mask

In [54]:
```
text = open('./An_Introduction_to_Fishing.txt', encoding='utf-8').read()
text = text.replace ('  ',' ')
wordcloud = WordCloud(width=800, height=400).generate(text)
#wordcloud = WordCloud(width=3600, height=1600).generate(text)

# Display the generated image:
plt.figure(figsize=(10,10)) # inches
plt.axis("off")
plt.imshow(wordcloud, interpolation='bilinear')
plt.show()
wordcloud = WordCloud(width=800, height=400,background_color='white').generate(
plt.figure(figsize=(10,10))
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show

mask = np.array(Image.open('./miloticMega.png'))

color= ImageColorGenerator(mask)

wordcloud = WordCloud(width=1920,
                      height=1080,
                      max_words=1000000000,
                      mask=mask,
                      stopwords=STOPWORDS,
                      background_color='white',
                      random_state=42).generate(text)

plt.figure(figsize=(20,20)) # inches
plt.axis("off")
plt.imshow(wordcloud.recolor(color_func=color),interpolation='bilinear')
plt.show()
```




fsh
fshing
water
hook
line
reel
rod
will
Fish
Fishing
lure
catch
bait
angler
often
may
Introduction
species
spool
cast
keep
time
lake
help
stream
release
trout
catfsh
hold
place
inches
near
good
bass
use
large
small
hand
cover
worm
food
target
bottom
drag
casting
pull
one
Many
eye
knot
Loop
end
make
plug


fsh
fshing
line
water
hook
lure
will
reel
cast
spool
place
bottom
lake
knot
food
good
rock
distance
Introduction
rod
inches
Don t
day
hold
scale
stream
often
bait
catfsh
Fishing
Fish
help
casting
start
time
large
drag
Many
surface
release
small
catch
keep
hand
angler
use
make
species
bass
size
usually
long
regulations
caught
end
live
worm
forward
feed
cover
may
grip
body
rod tip
one
back
habitat
trout
skin
Missouri
pull
found
along
point
daily
limit
head
others
aquatic
area
best
deep
rule
tie
sharp
remove
side
see
released
equipment
take
swim
cut
dock
small