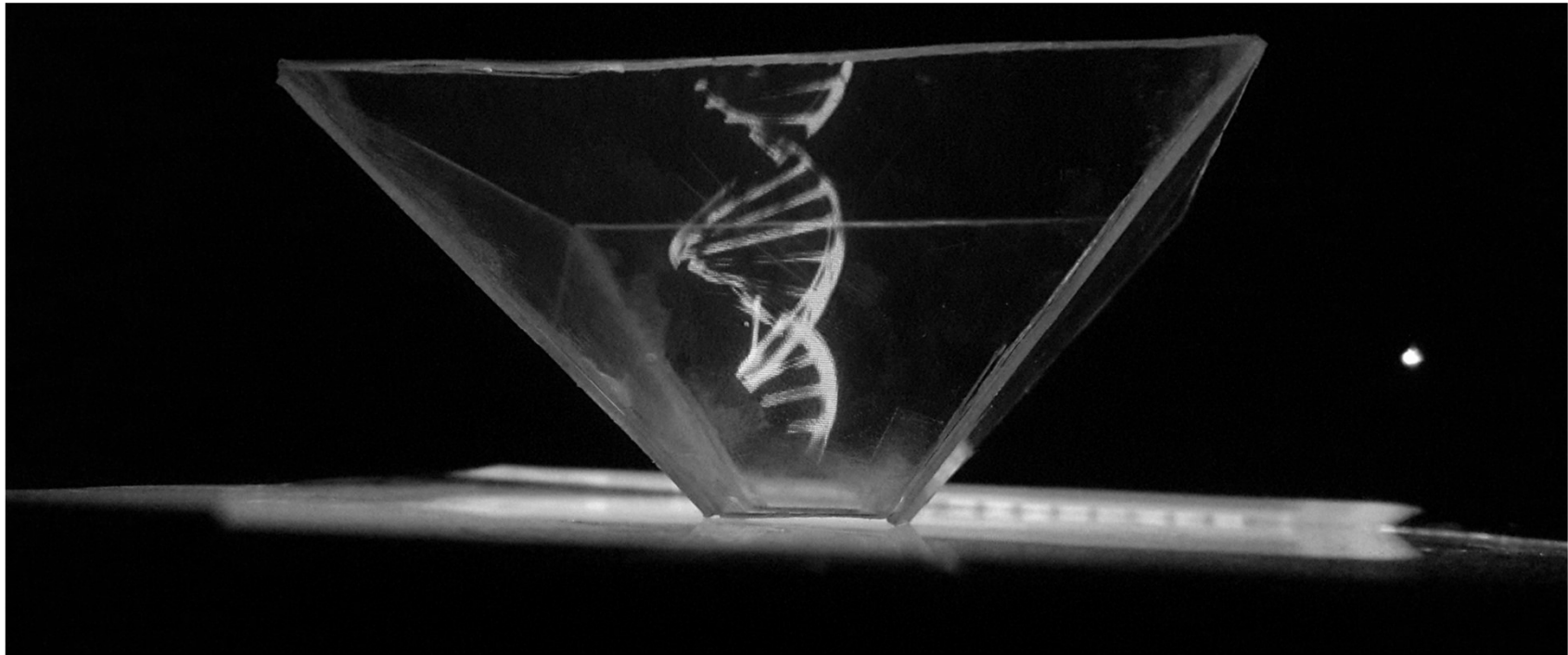


# 英和翻訳ホログラム



開発ソリューション部、開発センター  
Japar Jarkynbyek ジャハ

# 概要

## 1. テキスト読み上げ

Web Speech API

## 2. 英和翻訳

OpenNMT

PyTorch 機械学習トレーニング

## 3. ホログラフィックテキストを表示する

Python Flask

Html

JavaScript

CSS

## 4. ホログラフィックピラミッドディスプレイを作成する
















アクリルを使用して準備する

テキストの読み上げ処理

# Web Speech API

- 音声データをWebアプリに組み込む
- 使いやすいJavaScript API
- Web Platform Incubator Groupによって開発されました
- 「W3C Community Contributor」ライセンス契約に基づいて無料で使用できます
- 2つの部分があります
  - 音声合成（テキストから音声）
  - 音声認識（音声からテキストへ）

# Web Speech API ブラウザの互換性

SpeechRecognition												
												
	 Chrome	 Edge	 Firefox	 Internet Explorer	 Opera	 Safari	 Android webview	 Chrome for Android	 Firefox for Android	 Opera for Android	 Safari on iOS	 Samsung Internet
<div>SpeechRecognition</div> 	33 -x- *	?	No	No	No	No	Yes -x- *	Yes -x- *	No	No	No	Yes -x- *

-

 Full support
 

-

 No support

# ソースコードの例

```
1  try {  
2      var SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;  
3      var recognition = new SpeechRecognition();  
4  
5      $('#start-record-btn').on('click', function(e) {  
6          recognition.start();  
7      });  
8  
9      recognition.onstart = function() {  
10         instructions.text('Voice recognition activated. Try speaking into the microphone.');11     }  
12  
13     recognition.onspeechend = function() {  
14         instructions.text('You were quiet for a while so voice recognition turned itself off.');15     }  
16  
17     recognition.onerror = function(event) {  
18         if(event.error == 'no-speech') {  
19             instructions.text('No speech was detected. Try again.');20         }  
21     }  
22  
23     recognition.onresult = function(event) {  
24         var current = event.resultIndex;  
25  
26         // Get a transcript of what was said.  
27         var transcript = event.results[current][0].transcript;  
28  
29         // Add the current transcript to the contents of our Note.  
30         noteContent += transcript;  
31         noteTextarea.val(noteContent);  
32     }  
33 }  
34 catch(e) {  
35     console.error(e);  
36     $('#no-browser-support').show();  
37     $('#app').hide();  
38 }
```

## Speech to Text example

hello how are you

Start Recognition

Pause Recognition

# 英和翻訳の処理

# OpenNMT

OpenNMT は、ニューラル機械翻訳とニューラルシーケンス学習のためのオープンソースエコシステムです。

2017年にハーバードNLPグループとSYSTRANにより開始

いくつかの研究および産業アプリケーションで使用するプロジェクト



# 京都無料翻訳データ

京都に関するウィキペディアの記事に焦点を当てた日英翻訳

情報通信技術研究所が作成

日本語→英語のデータを翻訳し、プロの翻訳者がチェック

使用無料



# 情報処理

1. データをフォーマットに抽出する
2. データを単語にトークン化する  
英語のトークン化– Mosesツールキット  
日本語のトークン化– KyTea
3. データをトレーニングセットとテストセットに分離する
4. 1語未満または40語を超える文を削除するためのトレーニングデータのクリーニング

## 日本語テキストのフォーマットとトークン化

1 雪舟（せっしゅう、1420年（応永27年）-1506年（永正3年））は号で、15世紀  
2 日本の水墨画を一変させた。  
3 諱は「等楊（とうよう）」、もしくは「拙宗（せっしゅう）」と号した。  
4 備中国に生まれ、京都・相国寺に入ってから周防国に移る。  
5 その後遣明使に随行して中国（明）に渡って中国の水墨画を学んだ。  
6 作品は数多く、中国風の山水画だけでなく人物画や花鳥画もよくした。  
7 大胆な構図と力強い筆線は非常に個性的な画風を作り出している。  
8 現存する作品のうち6点が国宝に指定されており、日本の画家のなかでも別格の  
9 このため、花鳥図屏風などに「伝雪舟筆」される作品は大変多い。  
10 真筆であるか専門家の間でも意見の分かれるものも多々ある。



1 雪舟（せっしゅう、1420年（応永27年）-1506年（永正3年））は号  
2 日本の水墨画を一変させた。  
3 諱は「等楊（とうよう）」、もしくは「拙宗（せっしゅう）」と号した  
4 備中国に生まれ、京都・相国寺に入ってから周防国に移る。  
5 その後遣明使に随行して中国（明）に渡って中国の水墨画を学んだ  
6 作品は数多く、中国風の山水画だけでなく人物画や花鳥画もよくし  
7 大胆な構図と力強い筆線は非常に個性的な画風を作り出している。  
8 現存する作品のうち6点が国宝に指定されており、日本の画家のなか  
9 このため、花鳥図屏風などに「伝雪舟筆」される作品は大変多い。  
10 真筆であるか専門家の間でも意見の分かれるものも多々ある。

## 書式設定されトークン化された英語テキスト

1 Known as Sesshu (1420 - 1506), he was an ink painter and Zen monk active in th  
2 He revolutionized the Japanese ink painting.  
3 He was given the posthumous name "Toyo" or "Sesshu (拙宗)."  
4 Born in Bicchu Province, he moved to Suo Province after entering SShokoku-ji T  
5 Later he accompanied a mission to Ming Dynasty China and learned Chinese ink p  
6 His works were many, including not only Chinese-style landscape paintings, but  
7 His bold compositions and strong brush strokes constituted an extremely distin  
8 6 of his extant works are designated national treasures. Indeed, he is conside  
9 For this reason, there are a great many artworks that are attributed to him, s  
10 There are many works that even experts cannot agree if they are really his wor



1 Known as Sesshu ( 1420 - 1506 ) , he was an ink painter and Zen monk active in  
2 He revolutionized the Japanese ink painting .  
3 He was given the posthumous name " Toyo " or " Sesshu ( 拙宗 ) . "  
4 Born in Bicchu Province , he moved to Suo Province after entering SShokoku-ji  
5 Later he accompanied a mission to Ming Dynasty China and learned Chinese ink p  
6 His works were many , including not only Chinese-style landscape paintings , b  
7 His bold compositions and strong brush strokes constituted an extremely distin  
8 6 of his extant works are designated national treasures . Indeed , he is consi  
9 For this reason , there are a great many artworks that are attributed to him ,  
10 There are many works that even experts cannot agree if they are really his wor

# 情報処理

データ処理後のデータのサイズは次のとおりです

	Articles / 記事	Sentences / 文章	Japanese Words / 日本語言葉	English Words / 英語言葉
Train	14'126	440'000	12'000'000	11'500'000
Test	15	1'160	28'500	26'700

# トレーニング翻訳モデル

1. Anacondaをインストールする
2. Python 3.6の環境を作成する

```
C:¥>conda create --name translation python=3.6  
Collecting package metadata (current_repodata.json): done  
Solving environment: done  
C:¥>conda activate translation  
(translation) C:¥>
```

3. gitからOpenNMTソースコードを複製する

```
(translation) C:¥>cd work  
(translation) C:¥work>git clone https://github.com/OpenNMT/OpenNMT-py.git
```

4. PIPパッケージをインストールする

```
(translation) C:¥work>cd OpenNMT-py  
(translation) C:¥work¥OpenNMT-py>python setup.py install
```

## 5. データ準備を実行する

```
(translation) C:\work\OpenNMT-py>python preprocess.py -train_src data\japanese\kyoto-train.en -train_tgt data\japanese\kyoto-train.ja  
-valid_src data\japanese\kyoto-dev.en -valid_tgt data\japanese\kyoto-dev.ja -save_data data\en-ja
```

## 6. GPUでトレーニングデータを開始する

```
(translation) C:\work\OpenNMT-py>python train.py --gpu-ranks 0 -data data\en-ja -save_model data\en-ja-model
```

## 7. モデルを使用してテキストを翻訳する

```
C:\work\OpenNMT-py>python translate.py -model data\en-ja-model_step_100000.pt -src data\en-ja-model-test.txt
```

トレーニング後、トレーニング結果は70%でした

テキストをホログラ  
フィック形式で表示



# 使用技術

Python Flask

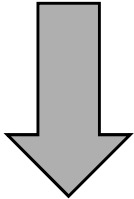
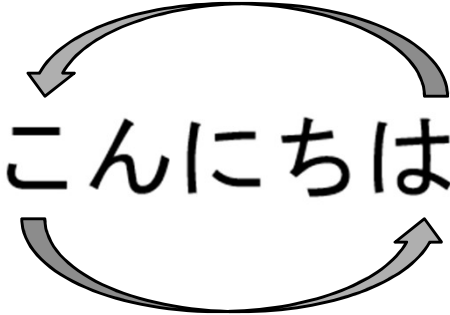
Html

JavaScript

CSS

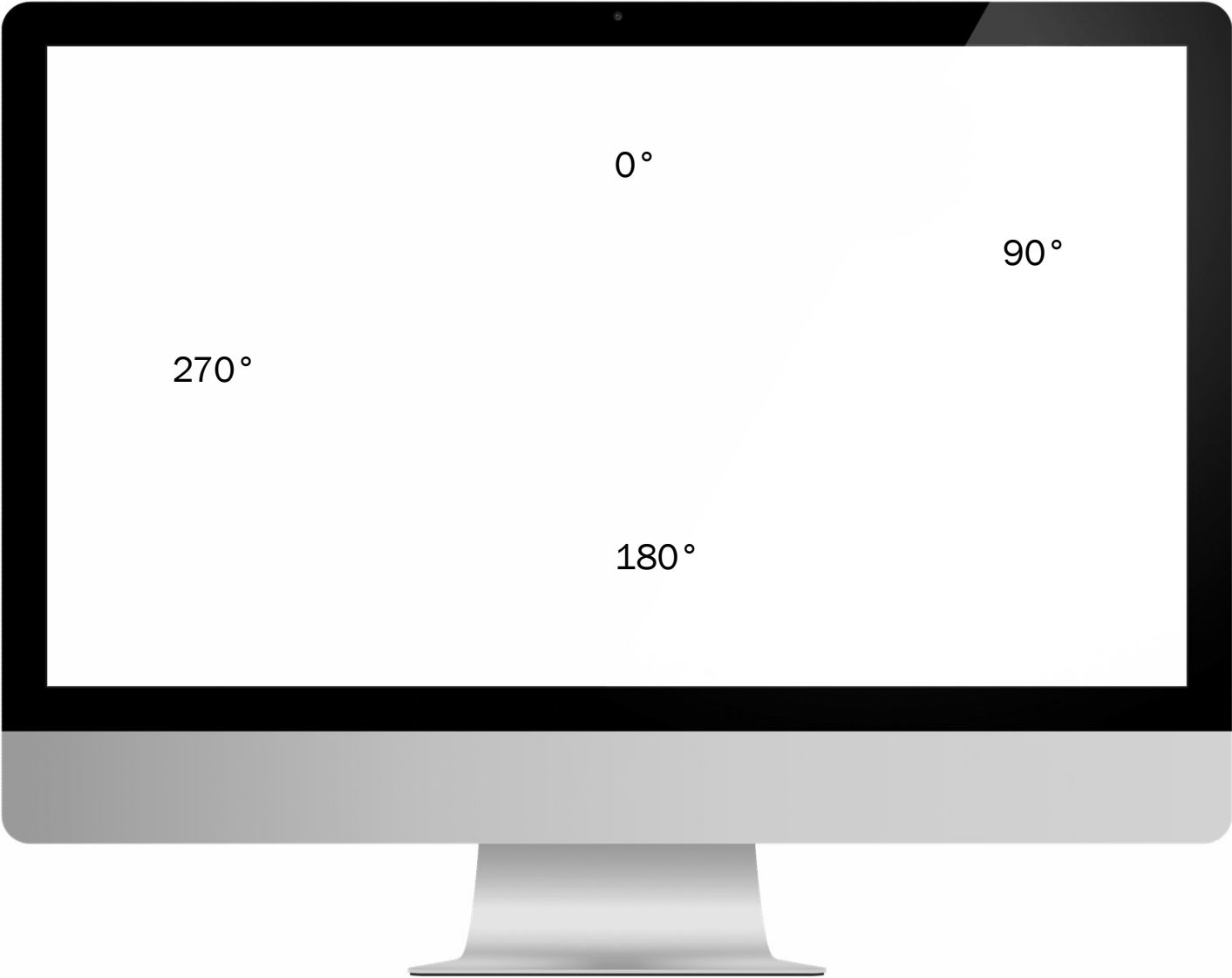


# ホログラム表示例



フリップ

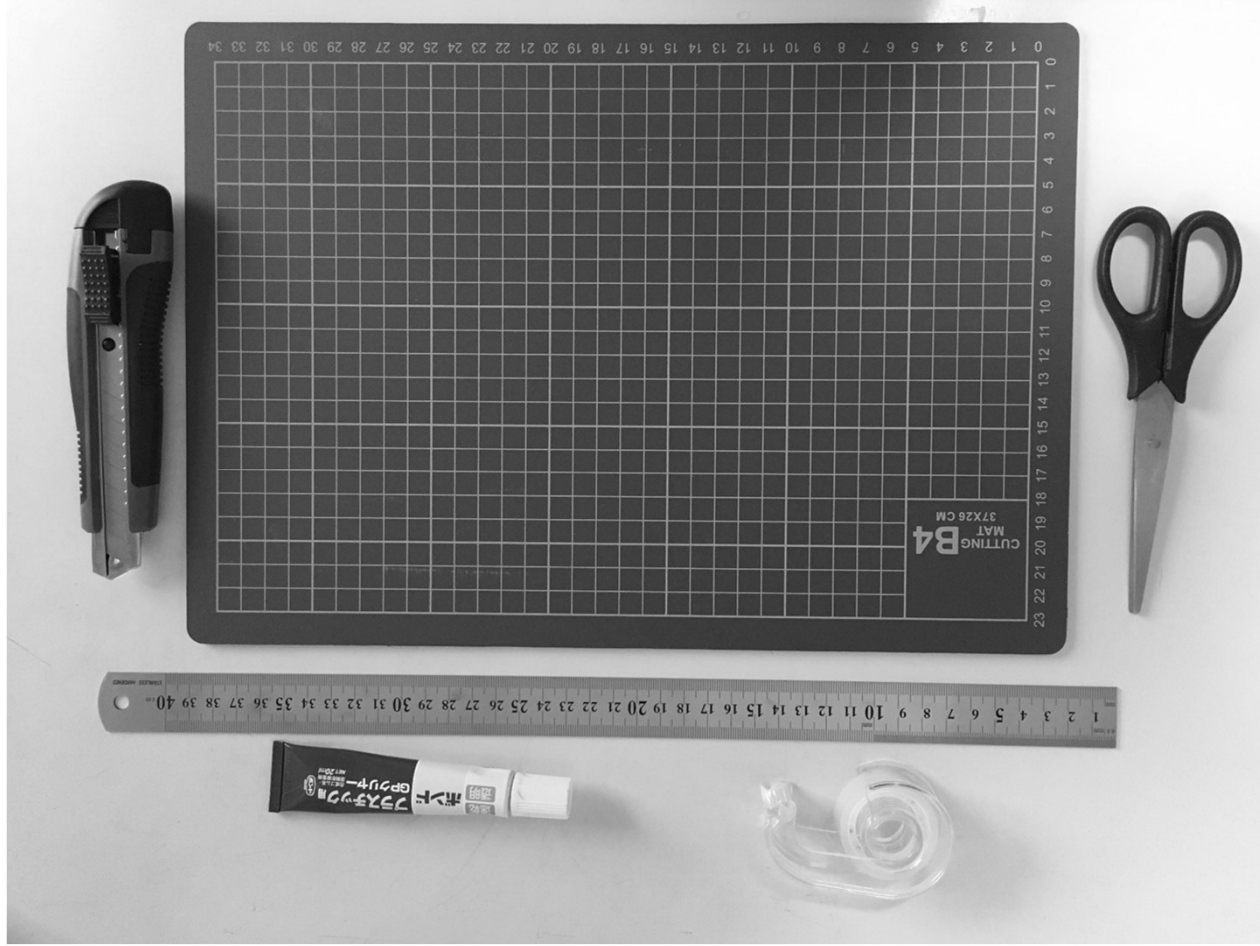
おさこいこ



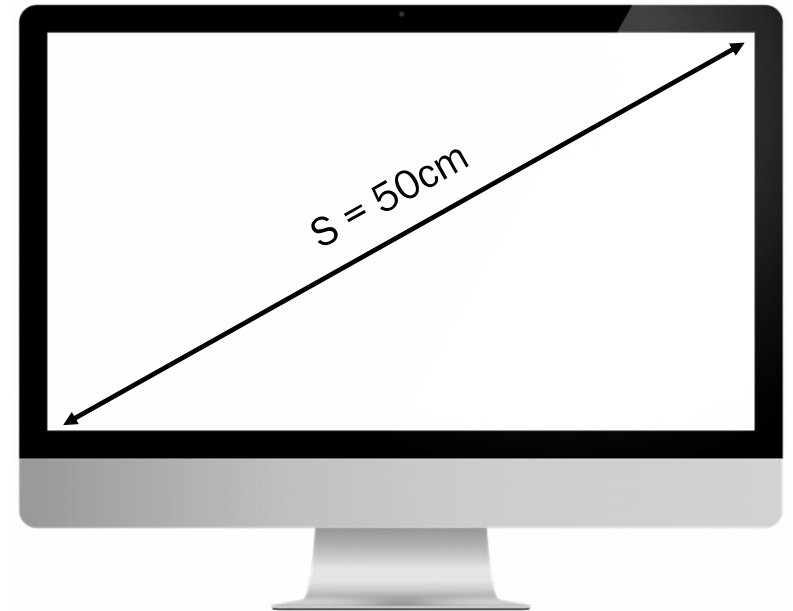
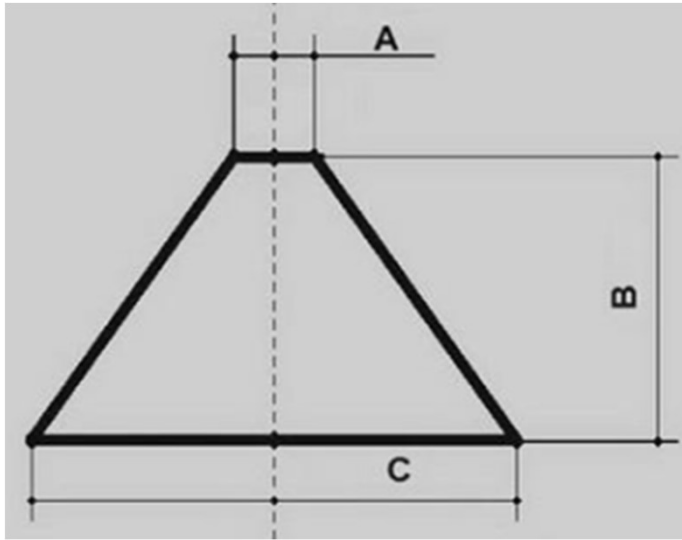


ホログラムピラミッドを作  
成する

# 準備

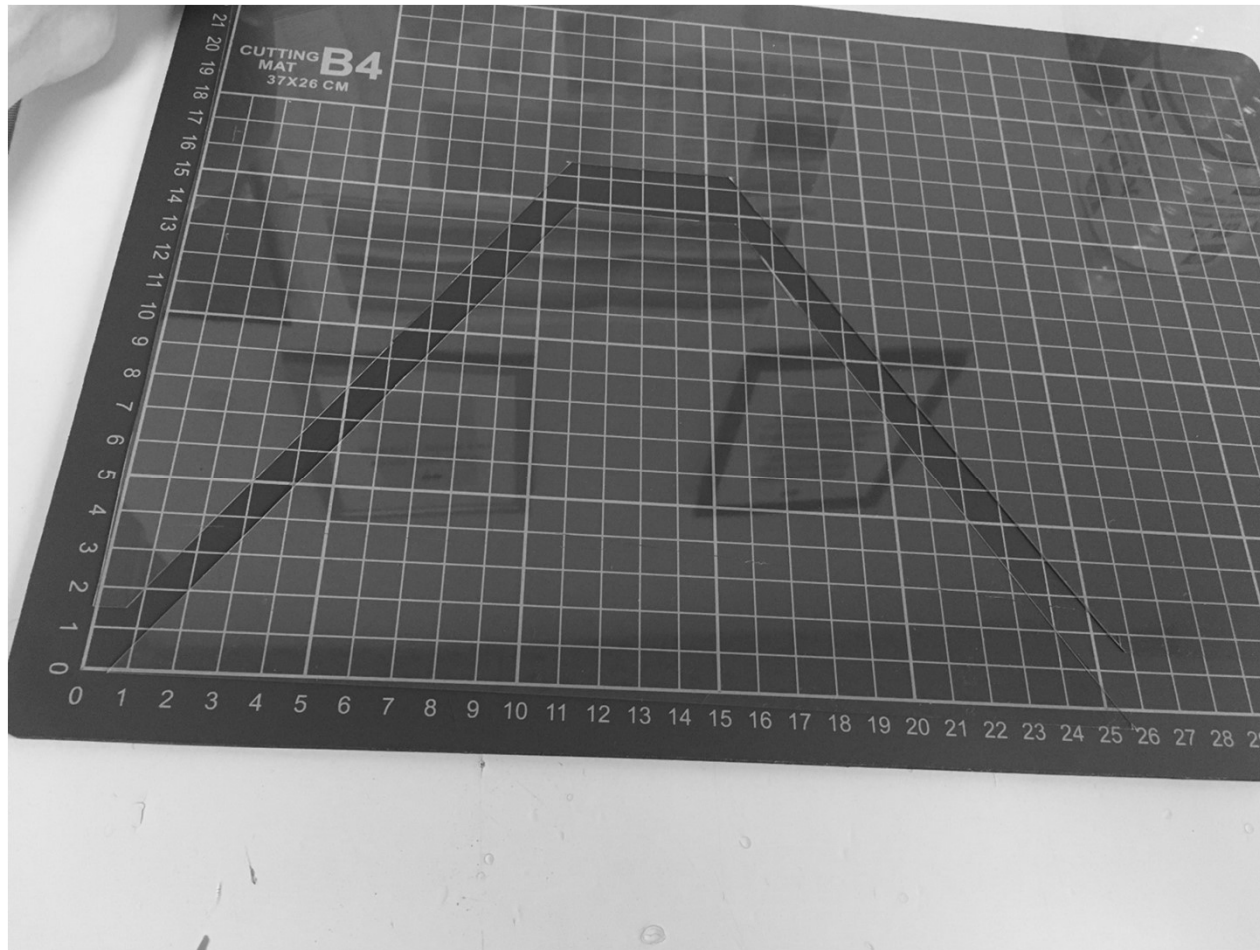


# 寸法を測定する

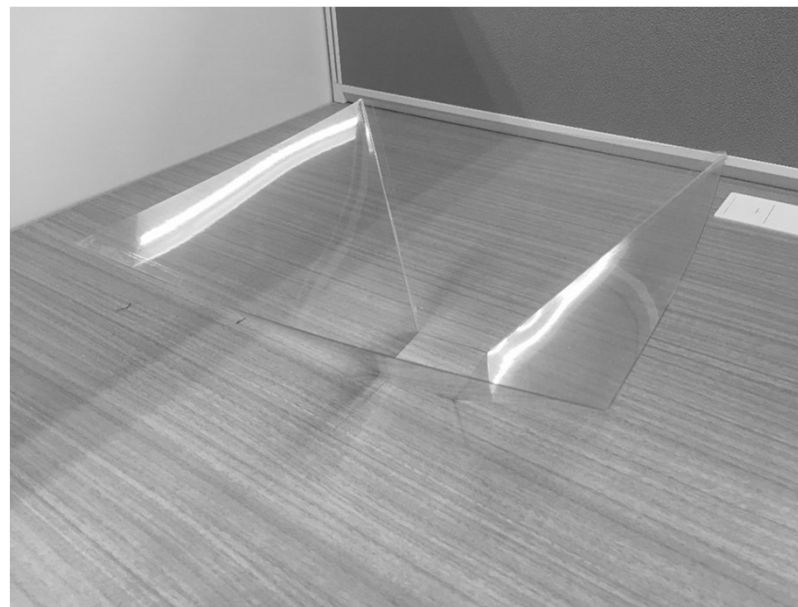
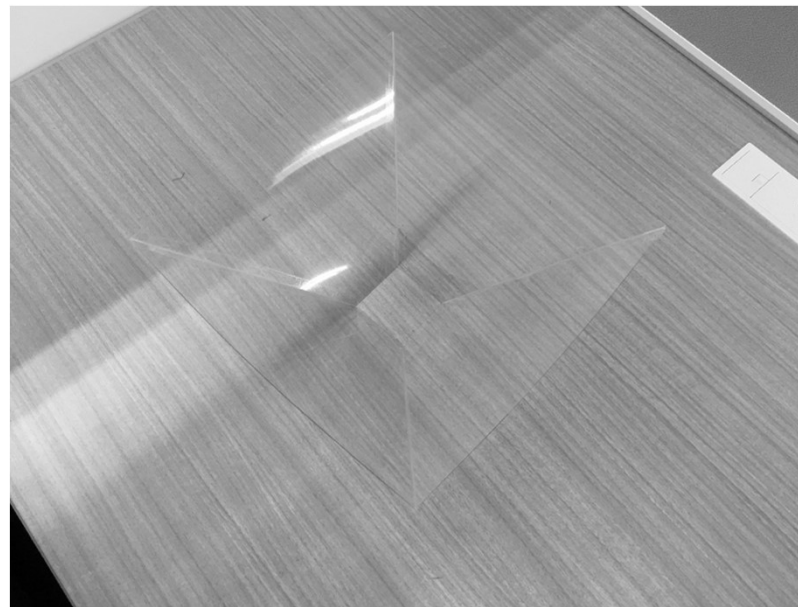
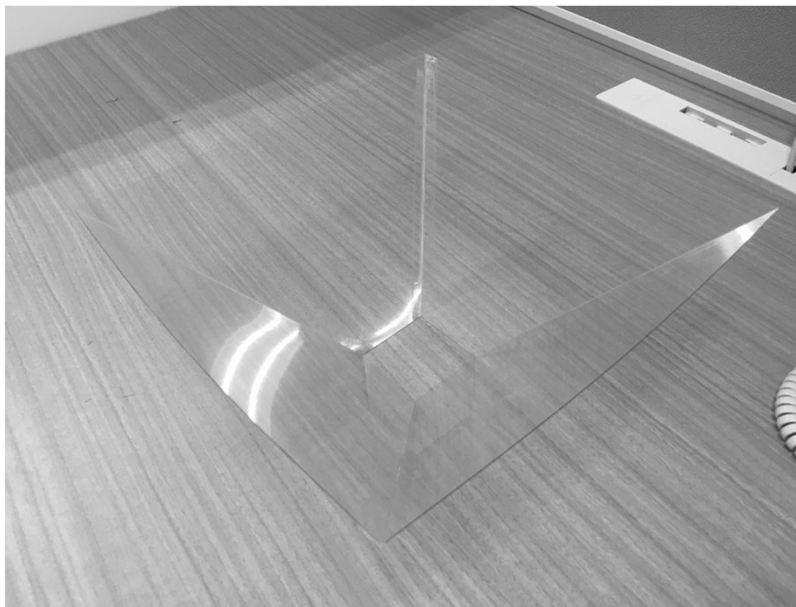


$$\begin{aligned} A &= (S / 5) * 0.5 = 5\text{cm} \\ B &= (S / 5) * 1.5 = 15\text{cm} \\ C &= (S / 5) * 2.5 = 25\text{cm} \end{aligned}$$

# アクリル4個カット



ホログラ  
フィック  
ピラミッド  
準備完了！





ご清聴ありがとうございました