

φυβλαςのblog

บล็อกของ phyblas (https://phyblas.hinaboshi.com)

ไทย
日本語 (https://phyblas.hinaboshi.com/nh)
中文 (https://phyblas.hinaboshi.com/zh)

หวัด

ถูกใจ

แชร์

ถูกใจ

แชร์

มีหนึ่งคนถูกใจสิ่งนี้ ถูกใจเป็นคนแรกในกลุ่มเพื่อนคุณเลิ

จัดการข้อมูลด้วย pandas เบื้องต้น บทที่ ๑๓: การเชื่อมตารางข้อมูล

เขียนเมื่อ 2016/09/25 15:39

แก้ไขล่าสุด 2020/06/02 19:57

ในบทที่แล้ว (https://phyblas.hinaboshi.com/yancham12) ได้เขียนถึงวิธีการเอาข้อมูลจากตารางสองอันขึ้นไปมารวมเข้าด้วยกันแล้ว สำหรับในบทนี้จะเป็นเรื่องของตารางสองตารางเข้าด้วยกันเช่นกันเพึ่งแต่ว่าเป็นการรวมในอีกลักษณะหนึ่ง นั่นคือการเชื่อมโดยผ่านความสัมพันธ์ของค่าภายในตาราง

การเชื่อมข้อมูลโดยผ่านคอลัมน์ที่ชื่อเหมือนกัน

หากมีตารางเตตาเฟรมสองอันแล้วต้องการนำข้อมูลมาเชื่อมต่อกันโดยอาศัยคอลัมน์หนึ่งเป็นตัวเชื่อมสามารถใช้ฟังก์ชัน pd.merge

ตัวอย่าง มีข้อมูลของโปเกมอนอยู่สองตาราง ตารางแรกบอกชนิดของโปเกมอน ส่วนอีกตารางบอกส่วนสูงและน้ำหนัก ทั้งสองตารางมีสองแถวแรกเป็นหมายเลขและสายพันธุ์ของโปเกมอนเหมือนกัน

```
import pandas as pd
pokemon_chanit = pd.DataFrame([
    [69, 'มาดัดลีโบมี', 'พีช/พีช'],
    [72, 'เมโนคราเงะ', 'น้ำ/พีช'],
    [74, 'อีซิดส์นึเดะ', 'หิน/ดิน']],
    columns=[ 'หมายเลข', 'สายพันธุ์', 'ชนิด'])
pokemon_sungnak = pd.DataFrame([
    [69, 'มาดัดลีโบมี', 0.7, 4.0],
    [72, 'เมโนคราเงะ', 0.9, 45.5],
    [74, 'อีซิดส์นึเดะ', 0.4, 20.0]],
    columns=[ 'หมายเลข', 'สายพันธุ์', 'ส่วนสูง', 'น้ำหนัก'])
print(pokemon_chanit)
print(pokemon_sungnak)
print(pd.merge(pokemon_chanit, pokemon_sungnak))
```



ได้

	หมายเลข	สายพันธุ์	ชนิด
0	69	มาดัดลีโบมี	พีช/พีช
1	72	เมโนคราเงะ	น้ำ/พีช
2	74	อีซิดส์นึเดะ	หิน/ดิน

	หมายเลข	สายพันธุ์	ส่วนสูง	น้ำหนัก
0	69	มาดัดลีโบมี	0.7	4.0
1	72	เมโนคราเงะ	0.9	45.5
2	74	อีซิดส์นึเดะ	0.4	20.0

	หมายเลข	สายพันธุ์	ชนิด	ส่วนสูง	น้ำหนัก
0	69	มาดัดลีโบมี	พีช/พีช	0.7	4.0

~ เกียวกับเรา ~
(/naenamtua)

สารบัญ

รวมคำแปลลิเด็ดจากญี่ปุ่น
(https://hinaboshi.com)

~ python ~ (/saraban/python)
มอดูลต่างๆ (/saraban/pythomoi
-- numpy
-- matplotlib
(/saraban/numpy_matplotlib)
-- pandas (/saraban/pandas)
-- opencv (/saraban/opencvpythc
-- pytorch (/saraban/pytorch)
การเรียนรู้ของเครื่อง
(/saraban/kanrianrukhangkhu
-- โครงข่าย
ประสาทเทียม
(/saraban/khong_khai_prasat_t

maya (/saraban/maya)

javascript (/saraban/javascript

ความน่าจะเป็น
(/saraban/khwamnachapen)

บันทึกในญี่ปุ่น (/saraban/nihon)

บันทึกในจีน (/saraban/zhonggu
-- บันทึกในปักกิ่ง (/saraban/beijing
-- บันทึกในฮ่องกง
(/saraban/hoenggong)
-- บันทึกในมาเก๊า
(/saraban/hoenggong)

บันทึกในไต้หวัน (/saraban/taiwa

บันทึกในยุโรปเหนือ
(/saraban/nordeuropa)

ได้

	สายพันธุ์	ชื่อเล่น	เลเวล
0	พูพริน	พูคุง	11
1	พิชู	พีโกะ	15
2	พิชู	พีลีเกะ	6
3	พูพริน	พูจิ้ง	3
4	โทเงปี	โทโทโระ	19
5	พิชู	พีนัด	7

	สายพันธุ์	ชนิด	ส่วนสูง	น้ำหนัก
0	พิชู	ไฟฟ้า	0.3	2.0
1	พี	ภูต	0.3	3.0
2	พูพริน	ธรรมชาติ/ภูต	0.3	1.0
3	โทเงปี	ภูต	0.3	1.5

จากนั้นก็ลองเอามารวมกัน

```
print(pd.merge(pokemon_thimi, khomun_pokemon))
```

ได้

	สายพันธุ์	ชื่อเล่น	เลเวล	ชนิด	ส่วนสูง	น้ำหนัก
0	พูพริน	พูคุง	11	ธรรมชาติ/ภูต	0.3	1.0
1	พูพริน	พูจิ้ง	3	ธรรมชาติ/ภูต	0.3	1.0
2	พิชู	พีโกะ	15	ไฟฟ้า	0.3	2.0
3	พิชู	พีลีเกะ	6	ไฟฟ้า	0.3	2.0
4	พิชู	พีนัด	7	ไฟฟ้า	0.3	2.0
5	โทเงปี	โทโทโระ	19	ภูต	0.3	1.5

จะเห็นว่าข้อมูลจากตารางอันหลังจะปรากฏซ้ำหลายตัวในตารางที่ได้จากการรวมนี้

ซึ่งความหมายในที่นี้ก็คือ สมมุติว่าตอนแรกเรารู้ว่าเรามีโปเกมอน ๖ ตัวนี้อยู่ รู้ว่าเป็นสายพันธุ์อะไร แต่ไม่รู้ว่าเป็นชนิดอะไรส่วนสูง น้ำหนักเท่าไร เราก็ต้องไปเปิดดูตารางข้อมูลโปเกมอนว่าสายพันธุ์นี้มีน้ำหนักเท่าไรสูงเท่าไรเป็นชนิดไหน

ในที่นี้พอรู้ว่าโปเกมอนตัวนั้นเป็นพิชู ไม่ว่ามันจะชื่อเล่นว่าอะไร เลเวลเท่าไร ถ้ามันเป็นพิชูมันก็ต้องเป็นชนิดไฟฟ้า สูง 0.3 น้ำหนัก 2.0 ดังนั้นข้อมูลตรงส่วนนี้จึงออกมาซ้ำกันทั้งหมด

ตรงจุดนี้เราจะเห็นได้ถึงประโยชน์ของการเชื่อมข้อมูล เพราะแทนที่เราจะต้องเก็บข้อมูลของพิชูแต่ละตัวซึ่งซ้ำกันทั้งหมด (ชนิด, ส่วนสูง, น้ำหนัก) เราเก็บแค่ที่ไม่ซ้ำกันไว้ในตารางหนึ่ง (ชื่อเล่น, เลเวล) แล้วเก็บที่ซ้ำไว้ในอีกตารางโดยเก็บแค่แถวเดียว จากนั้นพอต้องการก็ค่อยเอาตารางข้อมูลมาเชื่อมกัน แบบนี้ก็จะประหยัดพื้นที่เก็บข้อมูลได้มากมาย

อีกกรณีหนึ่งคือถ้าหากทั้งสองตารางต่างก็มีค่าของคอลัมน์นั้นซ้ำหลายค่าเหมือนกัน

```
chanit = pd.DataFrame([
    ['มิโนมูจิ', 'แมลง'],
    ['มิโนมาตาม', 'แมลง/พิช'],
    ['มิโนมาตาม', 'แมลง/ดิน'],
    ['มิโนมาตาม', 'แมลง/เหล็ก'],
    ['การ์เมล', 'แมลง/บิน']],
    columns=['สายพันธุ์', 'ชนิด'])
phalang = pd.DataFrame([
    ['มิโนมูจิ', 29, 45],
    ['มิโนมาตาม', 59, 85],
    ['มิโนมาตาม', 79, 105],
    ['มิโนมาตาม', 69, 95],
    ['การ์เมล', 94, 50]],
    columns=['สายพันธุ์', 'พลังโจมตี', 'พลังป้องกัน'])
print(chanit)
print(phalang)
```

หลักการเขียนทับศัพท์ภาษาจีนกลาง

日本語 (https://phyblas.hinaboshi.com/nh)

中文 (https://phyblas.hinaboshi.com/zh)

หลักการเขียนทับศัพท์ภาษาจีนกลาง
(/20181019)

g ในภาษาญี่ปุ่นออกเสียง "ก" หรือ "ง" กันแน่
(/20181017)

ทำความเข้าใจกับปัญญาประดิษฐ์และการเรียนรู้
ของเครื่อง (/panyapradit)

ค้นพบระบบดาวเคราะห์ ๘ ดวง เมืองหลังความ
สำเร็จคือปัญญาประดิษฐ์ (AI) (/20171216)

หอดูดาวโบราณปักกิ่ง ตอนที่ ๑: แทน
สังเกตการณ์และสวนดอกไม้ (/20170203)

พิพิธภัณฑ์สถาบันวิทยกรรมโบราณปักกิ่ง
(/20151028)

เที่ยวเมืองตานตง ล่องเรือในน่านน้ำ
เกาหลี่เหินโล (/20140927)

บันทึกการเที่ยวสวีเดน 1-12 พ.ค. 2014
(/20140514)

แนะนำองค์การวิจัยและพัฒนาการสำรวจ
อวกาศญี่ปุ่น (JAXA) (/20140116)

เล่าประสบการณ์ค่ายอบรมวิชาการทาง
ดาราศาสตร์โดยโซวเคนโด 10 - 16 พ.ย.
2013 (/20131205)

ตระเวนเที่ยวตามรอยจากของอนิเมะในญี่ปุ่น
(/20130820)

เที่ยวชมหอดูดาวที่ฐานสังเกตการณ์เซิงหลง
(/20130529)

บันทึกการเที่ยวญี่ปุ่นครั้งแรกในชีวิต - ทุกอย่าง
เริ่มต้นที่สนามบินนานาชาติคันไซ
(/20130118)

หลักการเขียนทับศัพท์ภาษาญี่ปุ่น
(/20130109)

ทำไมจึงไม่ควรเขียนวรรณยุกต์เวลาทับศัพท์
ภาษาต่างประเทศ (/20121010)

ทำไมถึงอยากมาเรียนต่อนอก (/20120622)

เหตุผลอะไรที่ต้องใช้ภาษาวิบัติ? (/20101106)

บทความแต่ละเดือน

2020年

1月 (/raiduean/202001) 2月 (/raiduean/202002) (/ra
5月 (/raiduean/202005) 6月 (/raiduean/202006) (/ra
9月 (/raiduean/202009) 10月 (/raiduean/202009)

2019年

1月 (/raiduean/201901) 2月 (/raiduean/201902) (/ra
5月 (/raiduean/201905) 6月 (/raiduean/201906) (/ra
9月 (/raiduean/201909) 10月 (/raiduean/201910) (/ra

2018年

1月 (/raiduean/201801) 2月 (/raiduean/201802) (/ra



ได้

	สายพันธุ์	ชนิด
0	มิโนมูจิ	แมลง
1	มิโนมาตาม	แมลง/พืช
2	มิโนมาตาม	แมลง/ดิน
3	มิโนมาตาม	แมลง/เหล็ก
4	การ์เมล	แมลง/บิน

	สายพันธุ์	พลังโจมตี	พลังป้องกัน
0	มิโนมูจิ	29	45
1	มิโนมาตาม	59	85
2	มิโนมาตาม	79	105
3	มิโนมาตาม	69	95
4	การ์เมล	94	50

ไปมอนที่มีมิโนมาตามมีอยู่ ๓ แบบ แต่ละแบบมีชนิดต่างกัน พลังโจมตีและพลังป้องกันก็ต่างกันด้วย ดังนั้นทั้ง ๒ ตารางนี้จึงมีมิโนมาตามอยู่ ๓ แถว

เมื่อลองมารวมกันดู

```
print(pd.merge(chanit, phalang))
```

จะได้

	สายพันธุ์	ชนิด	พลังโจมตี	พลังป้องกัน
0	มิโนมูจิ	แมลง	29	45
1	มิโนมาตาม	แมลง/พืช	59	85
2	มิโนมาตาม	แมลง/พืช	79	105
3	มิโนมาตาม	แมลง/พืช	69	95
4	มิโนมาตาม	แมลง/ดิน	59	85
5	มิโนมาตาม	แมลง/ดิน	79	105
6	มิโนมาตาม	แมลง/ดิน	69	95
7	มิโนมาตาม	แมลง/เหล็ก	59	85
8	มิโนมาตาม	แมลง/เหล็ก	79	105
9	มิโนมาตาม	แมลง/เหล็ก	69	95
10	การ์เมล	แมลง/บิน	94	50

ผลที่ได้พบว่ามิโนมาตาม ๙ แถว ซึ่งได้จากการแจกแจงตัวที่ซ้ำจากทั้งสองฝั่ง

ที่จริงเราจะต้องการให้ข้อมูลทั้ง ๓ แถวนั้นแนรวมกันโดยเรียงตามลำดับ แต่ในความเป็นจริง pd.merge จะไม่ได้มองแบบนั้น เมื่อเจอค่าที่ซ้ำกันมันไม่อาจจะได้ว่าเราต้องการให้แถวไหนเชื่อมกับแถวไหน ทำได้ก็เพียงแค่แจกแจงมันทั้งหมดเลย

หากต้องการจะแค่เอาคอลัมน์มาต่อกันจริงๆอาจลองใช้ concat จะเหมาะกว่า

```
print(pd.concat([chanit, phalang], axis=1))
```

ได้

	สายพันธุ์	ชนิด	สายพันธุ์	พลังโจมตี	พลังป้องกัน
0	มิโนมูจิ	แมลง	มิโนมูจิ	29	45
1	มิโนมาตาม	แมลง/พืช	มิโนมาตาม	59	85
2	มิโนมาตาม	แมลง/ดิน	มิโนมาตาม	79	105
3	มิโนมาตาม	แมลง/เหล็ก	มิโนมาตาม	69	95
4	การ์เมล	แมลง/บิน	การ์เมล	94	50

5月 6月

日本語 (https://phyblas.hinaboshi.com/nh)

中文 (https://phyblas.hinaboshi.com/zh)

2017年

1月 2月

5月 6月

9月 10月

2016年

1月 2月

5月 6月

9月 10月

ค้นบทความเก่ากว่านั้น (/raiduean)

	หมายเลข	สายพันธุ์	ชนิด	ส่วนสูง	น้ำหนัก
1	72	เมโนคราเงะ	น้ำ/พิษ	0.9	45.5
2	74	อิซิดีสึบะ	หิน/ดิน	0.4	20.0

จะเห็นว่าพอรวมกันแล้วได้ตารางที่มีชนิด, ส่วนสูง และน้ำหนัก โดยที่หมายเลขและสายพันธุ์ก็ยังอยู่และเหมือนเดิม ในที่นี้หมายเลขและสายพันธุ์ถูกใช้เป็นตัวเชื่อม

ปกติถ้าไม่ได้ระบุอะไรก็จะคอลัมน์ที่ซ้ำกันในทั้งสองตารางจะถูกใช้เป็นตัวเชื่อม แต่หากระบุไว้คอลัมน์ที่ถูกใช้เชื่อมก็จะมีแค่ตัวนั้น

การระบุตัวเชื่อมทำได้โดยใช้คีย์เวิร์ด on เช่นลอง

```
print(pd.merge(pokemon_chanit,pokemon_sungnak,on='สายพันธุ์'))
```

ได้

	หมายเลข_x	สายพันธุ์	ชนิด	หมายเลข_y	ส่วนสูง	น้ำหนัก
0	69	มาดดส์โบมี	พืช/พิษ	69	0.7	4.0
1	72	เมโนคราเงะ	น้ำ/พิษ	72	0.9	45.5
2	74	อิซิดีสึบะ	หิน/ดิน	74	0.4	20.0

จะเห็นว่าในที่นี้กำหนดให้ใช้แค่ "สายพันธุ์" เป็นตัวเชื่อม "หมายเลข" ไม่ได้เป็นตัวเชื่อมแล้วเลยถูกแยกออกเป็น ๒ อันแม้ว่าค่าจะเหมือนกันอยู่ก็ตาม โดยถูกเติมชื่อท้ายให้โดยอัตโนมัติ

ชื่อท้ายที่ถูกเติมสามารถกำหนดเองตามที่ต้องการได้โดยใช้คีย์เวิร์ด suffixes เป็นลิสต์ของสายอักขระที่ต้องการเติมต่อท้ายทั้งสองอัน เช่น

```
print(pd.merge(pokemon_chanit,pokemon_sungnak,on='สายพันธุ์',suffixes=[' ^_^','+_+']))
```

ได้

	หมายเลข ^_^	สายพันธุ์	ชนิด	หมายเลข +_+	ส่วนสูง	น้ำหนัก
0	69	มาดดส์โบมี	พืช/พิษ	69	0.7	4.0
1	72	เมโนคราเงะ	น้ำ/พิษ	72	0.9	45.5
2	74	อิซิดีสึบะ	หิน/ดิน	74	0.4	20.0

กรณีที่คอลัมน์ที่ใช้เชื่อมมีค่าซ้ำกัน

ความซับซ้อนมากขึ้นในการเชื่อมโยงตารางจะเกิดขึ้นเมื่อคอลัมน์ที่ใช้เชื่อมนั้นมีที่ค่าเหมือนกันสองตัวขึ้นไป กรณีแบบนี้ข้อมูลก็จะเกิดการซ้ำตามไปด้วย

ลองแยกดูเป็นกรณี เริ่มจากกรณีที่มีการซ้ำแค่ในตารางใดตารางหนึ่ง

ตัวอย่าง สร้างตารางขึ้นมาสองอัน ตารางแรกเป็นข้อมูลโปเกมอนที่มีอยู่ อีกตารางเป็นข้อมูลของโปเกมอนแต่ละสายพันธุ์

```
pokemon_thimi = pd.DataFrame([
    ['ทพุริน','พุด',11],
    ['พิช', 'พีโกะ',15],
    ['พิช', 'พีลีเกะ',6],
    ['ทพุริน','พูจิง',3],
    ['โทเงปี','โทโทโระ',19],
    ['พิช', 'พินัด',7]],
    columns=['สายพันธุ์','ชื่อเล่น','เลเวล'])
khomun_pokemon = pd.DataFrame([
    ['พิช','ไฟฟ้า',0.3,2.0],
    ['พี','ภูต',0.3,3.0],
    ['ทพุริน','ธรรมชาติ/ภูต',0.3,1.0],
    ['โทเงปี','ภูต',0.3,1.5]],
    columns=['สายพันธุ์','ชนิด','ส่วนสูง','น้ำหนัก'])
print(pokemon_thimi)
print(khomun_pokemon)
```



บันทึกในไซต์อื่น

日本語 ([/saraban/langdabon](https://phyblas.hinaboshi.com/nh))

中文 (<https://phyblas.hinaboshi.com/zh>)

เรียนภาษาจีน ([/saraban/gakken](https://phyblas.hinaboshi.com/zh))

qiita ([/qiita](https://qiita.com))

บทความอื่นๆ ([/saraban/uenuen](https://phyblas.hinaboshi.com))

บทความแบ่งตามหมวด (/muat)

==เลือกหมวด==

ติดตามอัปเดตของบล็อกได้ที่แฟนเพจ



ค้นหาบทความ

ค้น

บทความล่าสุด

ความน่าจะเป็นเบื้องต้นสำหรับเขียนโปรแกรม บทที่ ๒๒: การแจกแจงดิสทริเบล (/gasha22)

ความน่าจะเป็นเบื้องต้นสำหรับเขียนโปรแกรม บทที่ ๒๑: การแจกแจงอเนกนามและการแจกแจงแบบหมวดหมู่ (/gasha21)

ความน่าจะเป็นเบื้องต้นสำหรับเขียนโปรแกรม บทที่ ๒๐: การแจกแจงสวิตเดนดท์ (/gasha20)

ความน่าจะเป็นเบื้องต้นสำหรับเขียนโปรแกรม บทที่ ๑๙: การคาดการณ์ค่าที่พหามิเตอร์การแจกแจงมีความไม่แน่นอน (/gasha19)

ความน่าจะเป็นเบื้องต้นสำหรับเขียนโปรแกรม บทที่ ๑๘: การแจกแจงโคกาสังสอง (/gasha18)

บทความแนะนำ

ภาษาจีนแบ่งเป็นสำเนียงอะไรบ้าง มีความแตกต่างกันมากแค่ไหน (/20200520)

ทำความเข้าใจระบอบประชาธิปไตยจากประวัติศาสตร์ความเป็นมา (/prawatsat_prachathipatai)

เรียนรู้การใช้ regular expression (regex) (/20190709)

ไทย

日本語 (<https://phyblas.hinaboshi.com/nh>)中文 (<https://phyblas.hinaboshi.com/zh>)

รูปแบบของการเชื่อมโยง

จากตัวอย่างที่ผ่านมา ข้อมูลในคอลัมน์ที่ใช้เชื่อมจากทั้ง ๒ ตารางนั้นส่วนใหญ่ต่างมีค่าเหมือนกันหมด แต่ในความเป็นจริงไม่จำเป็นต้องเป็นเช่นนั้น ในตารางหนึ่งอาจมีค่าที่ในอีกตารางไม่มีอยู่

ตัวอย่าง มีตารางข้อมูลโปเกมอนสองตาราง อันหนึ่งให้ข้อมูลส่วนสูงและน้ำหนัก อีกอันเป็นข้อมูลค่าพลังต่างๆ ทั้งสองอันนี้มีสมาชิกบางส่วนไม่เหมือนกัน

```
sungnak = pd.DataFrame([
    ['โชนาโน', 0.6, 14.0],
    ['โชนานัส', 1.3, 28.5],
    ['ยุกิวาราช', 0.7, 16.8],
    ['โอนิโกริ', 1.5, 256.5]],
    columns=['สายพันธุ์', 'ส่วนสูง', 'น้ำหนัก'])
phalang = pd.DataFrame([
    ['โชนาโน', 23, 48, 95],
    ['ยุกิวาราช', 50, 50, 50],
    ['โอนิโกริ', 80, 80, 80],
    ['ยูกิเมะโนะโกะ', 70, 80, 70]],
    columns=['สายพันธุ์', 'พลังโจมตี', 'พลังป้องกัน', 'HP'])
print(sungnak)
print(phalang)
```



ได้

	สายพันธุ์	ส่วนสูง	น้ำหนัก
0	โชนาโน	0.6	14.0
1	โชนานัส	1.3	28.5
2	ยุกิวาราช	0.7	16.8
3	โอนิโกริ	1.5	256.5

	สายพันธุ์	พลังโจมตี	พลังป้องกัน	HP
0	โชนาโน	23	48	95
1	ยุกิวาราช	50	50	50
2	โอนิโกริ	80	80	80
3	ยูกิเมะโนะโกะ	70	80	70

ลองเอามาเชื่อมกัน

```
print(pd.merge(sungnak, phalang))
```

ผลที่ได้

	สายพันธุ์	ส่วนสูง	น้ำหนัก	พลังโจมตี	พลังป้องกัน	HP
0	โชนาโน	0.6	14.0	23	48	95
1	ยุกิวาราช	0.7	16.8	50	50	50
2	โอนิโกริ	1.5	256.5	80	80	80

จะเห็นว่าข้อมูลที่ไม่ได้มีอยู่ในทั้งสองตารางเหมือนกันจะหายไปเลย ทำให้เหลือแค่ ๓ แถว

ทั้งนี้เป็นเพราะว่าโดยทั่วไปแล้วการเชื่อมโยงตารางถูกตั้งค่าเริ่มต้นให้เป็นแบบ inner นั่นคือเก็บเฉพาะที่มีเหมือนกันทั้งสองตาราง

แต่นอกจากนี้แล้วยังมีอีก ๓ แบบ คือ

left เก็บที่ตัวซ้ายมีทั้งหมดโดยไม่สนตัวขวา

right เก็บที่ตัวขวามีทั้งหมดโดยไม่สนตัวซ้าย

outer เก็บทั้งที่ตัวซ้ายมีและตัวขวามี

สามารถกำหนดว่าจะเชื่อมโยงแบบไหนโดยตั้งที่คีย์เวิร์ด how โดยที่ถ้าไม่ใส่จะเท่ากับ how='inner'

ตัวอย่าง

```
print(pd.merge(sungnak, phalang, how= 'outer' ))
```

ไทย

日本語 (<https://phyblas.hinaboshi.com/nh>)

中文 (<https://phyblas.hinaboshi.com/zh>)

ได้

	สายพันธุ์	ส่วนสูง	น้ำหนัก	พลังโจมตี	พลังป้องกัน	HP
0	โซนาโน	0.6	14.0	23.0	48.0	95.0
1	โซแนนส์	1.3	28.5	NaN	NaN	NaN
2	ยูกิวาราชิ	0.7	16.8	50.0	50.0	50.0
3	โอนิโกริ	1.5	256.5	80.0	80.0	80.0
4	ยูกิเมะโนะโกะ	NaN	NaN	70.0	80.0	70.0

จะเห็นว่าข้อมูลที่มีอยู่แค่ฝั่งใดฝั่งหนึ่งก็มาโผล่ในนี้ด้วย แต่ส่วนที่ขาดไปก็จะขึ้นเป็น NaN

นอกจากนี้หากลอง left กับ right ก็ออกมาในทำนองเดียวกัน

```
print('ซ้าย')
print(pd.merge(sungnak, phalang, how= 'left' ))
print('ขวา')
print(pd.merge(sungnak, phalang, how= 'right' ))
```

ได้

ซ้าย

	สายพันธุ์	ส่วนสูง	น้ำหนัก	พลังโจมตี	พลังป้องกัน	HP
0	โซนาโน	0.6	14.0	23.0	48.0	95.0
1	โซแนนส์	1.3	28.5	NaN	NaN	NaN
2	ยูกิวาราชิ	0.7	16.8	50.0	50.0	50.0
3	โอนิโกริ	1.5	256.5	80.0	80.0	80.0

ขวา

	สายพันธุ์	ส่วนสูง	น้ำหนัก	พลังโจมตี	พลังป้องกัน	HP
0	โซนาโน	0.6	14.0	23	48	95
1	ยูกิวาราชิ	0.7	16.8	50	50	50
2	โอนิโกริ	1.5	256.5	80	80	80
3	ยูกิเมะโนะโกะ	NaN	NaN	70	80	70

เมื่อชื่อคอลัมน์ที่ต้องการเชื่อมไม่เหมือนกันในสองตาราง

ไม่ว่าจะสละคอลัมน์จากสองตารางที่เอามาเชื่อมกันจะมีค่าชื่อคอลัมน์เหมือนกันเสมอไป ในกรณีที่แตกต่างกันแบบนี้จะต้องกำหนดชื่อคอลัมน์ที่ต้องการเชื่อมโดยใช้ดีวีเวิร์ด left_on=ชื่อคอลัมน์ในตารางซ้าย และ right_on=ชื่อคอลัมน์ในตารางขวา

ตัวอย่าง

```
chue = pd.DataFrame([
    [480, 'ยูกชี'],
    [481, 'เฮอร์ริต'],
    [482, 'อัลโนม']],
    columns=[ 'เลข', 'สายพันธุ์' ])
phalang = pd.DataFrame([
    [480, 75, 130],
    [481, 105, 105],
    [482, 125, 70]],
    columns=[ 'หมายเลข', 'พลังโจมตี', 'พลังป้องกัน' ])
print(chue)
print(phalang)
```



ได้

	เลข	สายพันธุ์
0	480	ยูกชี

	เลข	สายพันธุ์	ไทย
1	481	เอมริด	日本語 (https://phyblas.hinaboshi.com/nh)
2	482	อักนอม	中文 (https://phyblas.hinaboshi.com/zh)

	หมายเลข	พลังโจมตี	พลังป้องกัน
0	480	75	130
1	481	105	105
2	482	125	70

ในที่นี้จะเชื่อมสองตารางเข้าใช้ "เลข" ตารางขวาใช้ "หมายเลข"

```
print(pd.merge(chue, phalang, left_on='เลข', right_on='หมายเลข'))
```

ผลที่ได้จะเห็นทั้งสองคอลัมน์อยู่พร้อมกันโดยมีค่าเท่ากัน

	เลข	สายพันธุ์	หมายเลข	พลังโจมตี	พลังป้องกัน
0	480	ยุกชี	480	75	130
1	481	เอมริด	481	105	105
2	482	อักนอม	482	125	70

การใช้ดัชนีเป็นตัวเชื่อม

ปกติแล้ว merge จะใช้คอลัมน์อันหนึ่งหรือจำนวนหนึ่งเป็นตัวเชื่อมโยงข้อมูลสองตารางเข้าด้วยกัน โดยที่ค่าดัชนีจะถูกเพิกเฉยและโยนทิ้งไปเลย

แต่หากต้องการใช้ดัชนีมาเป็นตัวเชื่อมก็ทำได้เช่นกัน โดยจะต้องใส่คีย์เวิร์ด left_index=1 หากต้องการให้ตัวซ้ายใช้ดัชนี และ right_index=1 หากต้องการให้ตัวขวาใช้ดัชนี

ส่วนอีกตัวที่ไม่ได้ใช้ดัชนีก็ให้กำหนดคอลัมน์ที่ต้องการเชื่อมด้วย left_on และ right_on

ตัวอย่าง

```
pokemon = pd.DataFrame([
    ['ชิโคริดา', 'พืช'], ['อีโนอาราชิ', 'ไฟ'], ['วานินโนโกะ', 'น้ำ'],
    ['คิโมริ', 'พืช'], ['อาซาโม', 'ไฟ'], ['มิซึโงโรว', 'น้ำ']],
    columns=['สายพันธุ์', 'ชนิด'],
    index=[152, 155, 158, 252, 255, 258])
phaechana = pd.DataFrame([
    ['เสมอ', 'แพ', 'ชนะ'],
    ['ชนะ', 'เสมอ', 'แพ'],
    ['แพ', 'ชนะ', 'เสมอ'],
    ['แพ', 'ชนะ', 'แพ']],
    columns=['เจอพืช', 'เจอไฟ', 'เจอน้ำ'],
    index=['พืช', 'ไฟ', 'น้ำ', 'ดิน'])
print(pokemon)
print(phaechana)
print(pd.merge(pokemon, phaechana, left_on='ชนิด', right_index=1))
```



ได้

	สายพันธุ์	ชนิด
152	ชิโคริดา	พืช
155	อีโนอาราชิ	ไฟ
158	วานินโนโกะ	น้ำ
252	คิโมริ	พืช
255	อาซาโม	ไฟ
258	มิซึโงโรว	น้ำ

	เจอพืช	เจอไฟ	เจอน้ำ
พืช	เสมอ	แพ	ชนะ

	เจอพืช	เจอไฟ	เจอน้ำ	ไทย
ไฟ	ชนะ	เสมอ	แพ้	日本語 (https://phyblas.hinaboshi.com/nh)
น้ำ	แพ้	ชนะ	เสมอ	中文 (https://phyblas.hinaboshi.com/zh)
ดิน	แพ้	ชนะ	แพ้	

	สายพันธุ์	ชนิด	เจอพืช	เจอไฟ	เจอน้ำ
152	ชิโครีตา	พืช	เสมอ	แพ้	ชนะ
252	คิโมริ	พืช	เสมอ	แพ้	ชนะ
155	ฮิโนอาราชิ	ไฟ	ชนะ	เสมอ	แพ้
255	อาซาโม	ไฟ	ชนะ	เสมอ	แพ้
158	วานิโนโกะ	น้ำ	แพ้	ชนะ	เสมอ
258	มิชิโงโรว	น้ำ	แพ้	ชนะ	เสมอ

แต่หากใช้ left_index=1 และ right_index=1 พร้อมกันผลที่ได้จะไม่ต่างจาก concat

สรุปเค้เวิร์ดทั้งหมด		
เค้เวิร์ด	ความหมาย	ค่าตั้งต้น
how	วิธีการในการรวม	inner
on	ชื่อคอลัมน์ที่ใช้เป็นตัวเชื่อม ไม่ใช่ใช้พร้อมกับ left_on และ right_on	คอลัมน์ที่ชื่อซ้ำกันทั้งหมด
left_on	ชื่อคอลัมน์ของตารางซ้ายที่ใช้เป็นตัวเชื่อม ไม่ใช่ใช้พร้อมกับ on	
right_on	ชื่อคอลัมน์ของตารางขวาที่ใช้เป็นตัวเชื่อม ไม่ใช่ใช้พร้อมกับ on	
left_index	ใช้ดัชนีของตารางซ้ายเป็นตัวเชื่อม	0
right_index	ใช้ดัชนีของตารางขวาเป็นตัวเชื่อม	0
sort	จัดเรียงแถวในตารางตามค่าดัชนีด้วย หลังจากที่ได้เชื่อมเสร็จ	0
suffixes	คำต่อท้ายหากชื่อคอลัมน์ที่ไม่ใช่ตัวเชื่อมเกิดซ้ำกัน	[_x,_y]

การใช้เมธอด join

นอกจากจะใช้ฟังก์ชัน pd.merge ในการเชื่อมตารางแล้ว ยังอาจใช้เมธอด join ของตัวเตดาเฟรมได้ วิธีการใช้ใกล้เคียงกัน เพียงแต่จะใช้เตดาเฟรมตัวหนึ่งเป็นฐานของเมธอด แล้วใช้อีกอันเป็นอาร์กิวเมนต์

join ยังต่างจาก pd.merge ตรงที่ว่าหากไม่ระบุเค้เวิร์ด on จะเป็นการเชื่อมโดยใช้ดัชนีเป็นตัวเชื่อม ต่างจาก pd.merge ที่ใช้คอลัมน์ที่ซ้ำ

ตัวอย่าง

```
sungnak = pd.DataFrame([
    ['เตียลกา', 5.4, 683.0],
    ['พาลเกีย', 4.2, 336.0],
    ['กิราทีนา', 4.5, 750.0]],
    columns=['สายพันธุ์', 'ส่วนสูง', 'น้ำหนัก'],
    index=[483, 484, 487])
phalang = pd.DataFrame([
    [120, 120], [120, 100], [100, 120]],
    columns=['พลังโจมตี', 'พลังป้องกัน'],
    index=[483, 484, 487])
print(sungnak)
print(phalang)
print(sungnak.join(phalang))
```



ได้

	สายพันธุ์	ส่วนสูง	น้ำหนัก
483	เตียลกา	5.4	683.0
484	พาลเกีย	4.2	336.0
487	กิราทีนา	4.5	750.0

	พลังโจมตี	พลังป้องกัน
483	120	120
484	120	100
487	100	120

ไทย

日本語 (<https://phyblas.hinaboshi.com/nh>)中文 (<https://phyblas.hinaboshi.com/zh>)

	สายพันธุ์	ส่วนสูง	น้ำหนัก	พลังโจมตี	พลังป้องกัน
483	เตี้ยลกา	5.4	683.0	120	120
484	พาลเกีย	4.2	336.0	120	100
487	กิราทีนา	4.5	750.0	100	120

ถ้าหากใส่คีย์เวิร์ด on ลงไปจะเป็นการระบุคอลัมน์ของเตตาเฟรมที่เป็นเจ้าของเมรอด ส่วนเตตาเฟรมที่มาเดิมนั้นยังไ้ก็ต้องเดิมด้วย คำนี้นี้ ไม่สามารถเชื่อมตามคอลัมน์ได้

ตัวอย่าง ถ้าเรากำหนดเตตาเฟรมตัวที่มาเดิมนั้นเป็น

```
phalang2 = pd.DataFrame([
    [120,120],[120,100],[100,120]],
    columns=[ 'พลังโจมตี', 'พลังป้องกัน' ],
    index=[ 'เตี้ยลกา', 'พาลเกีย', 'กิราทีนา' ])
print(phalang2)
```

ได้

	พลังโจมตี	พลังป้องกัน
เตี้ยลกา	120	120
พาลเกีย	120	100
กิราทีนา	100	120

ต่างจากตัวอย่างที่แล้วตรงที่ว่า index ไม่ใช่หมายเลขแต่กลายเป็นชื่อสายพันธุ์ของโปเกมอน แบบนี้จะต้องพิมพ์

```
print(sungnak.join(phalang2,on='สายพันธุ์'))
```

ก็จะได้ผลเหมือนกับตัวอย่างที่แล้ว

และ join ไม่ได้ใช้แค่กับเตตาเฟรมแต่ยังใช้กับซีรีส์ได้ด้วย โดยจะเป็นการเพิ่มคอลัมน์ให้กับเตตาเฟรม อีกทั้งยังใส่หลายตัวพร้อมกันได้ ด้วย โดยต้องใส่เป็นลิสต์ของเตตาเฟรมหรือซีรีส์ที่ต้องการเพิ่ม

ตัวอย่าง ลองสร้างซีรีส์ที่เก็บชนิดของโปเกมอน

```
chanit = pd.Series(['เหล็ก/มังกร', 'น้ำ/มังกร', 'ไฟ/มังกร'],
    index=[483,484,487],name='ชนิด')
print(chanit)
```

ได้

```
483    เหล็ก/มังกร
484     น้ำ/มังกร
487     ไฟ/มังกร
Name: ชนิด, dtype: object
```

จากนั้นก็นำมาใช้เชื่อมกับข้อมูลของสองตารางที่สร้างขึ้นมาก่อนหน้าด้วย เป็นการเชื่อมข้อมูล ๓ ตารางในที่เดียว

```
print(phalang.join([chanit,sungnak]))
```

ได้

	พลังโจมตี	พลังป้องกัน	ชนิด	สายพันธุ์	ส่วนสูง	น้ำหนัก
483	120	120	เหล็ก/มังกร	เตี้ยลกา	5.4	683.0
484	120	100	น้ำ/มังกร	พาลเกีย	4.2	336.0
487	100	120	ไฟ/มังกร	กิราทีนา	4.5	750.0

สามารถใช้คีย์เวิร์ด how เพื่อกำหนดรูปแบบการเชื่อมโยงได้ เพียงแต่ถ้าหากไม่กำหนดไว้ค่าเริ่มต้นจะเป็น left นั่นคือยึดเอาเตตาเฟรมที่เป็นฐานของเมรอดเป็นหลัก

นอกจากนี้ยังมีข้อแตกต่างเล็กน้อยอีกเช่น คีย์เวิร์ด suffixes ใน pd.merge นั้นจะเทียบเท่ากับคีย์เวิร์ด lsuffix กับ rsuffix ของ join กล่าวคือ join ใช้คีย์เวิร์ดแยก ๒ ตัวแทนที่จะรวมกันเป็นลิสต์

ตัวอย่าง

```
pokemon1 = pd.DataFrame([
    ['มานิวลา', 1.1, 34.0, 'มาร'],
    ['จินะคอยล์', 1.2, 180.0, 'ไฟฟ้า'],
    ['เบโรเบลต์', 1.7, 140.0, 'ธรรมชาติ'],
    ['โดไซดอน', 2.4, 282.8, 'ดิน']],
    columns=['สายพันธุ์', 'ส่วนสูง', 'น้ำหนัก', 'ชนิด'],
    index=[461, 462, 463, 464])
pokemon2 = pd.DataFrame([
    ['น้ำแข็ง', 120, 65],
    ['เหล็ก', 70, 115],
    ['หิน', 140, 130],
    ['พืช', 100, 125]],
    columns=['ชนิด', 'พลังโจมตี', 'พลังป้องกัน'],
    index=[461, 462, 464, 465])
print(pokemon1)
print(pokemon2)
print(pokemon1.join(pokemon2, lsuffix=' 1', rsuffix=' 2'))
```

ไทย

日本語 (<https://phyblas.hinaboshi.com/nh>)

中文 (<https://phyblas.hinaboshi.com/zh>)



ได้

	สายพันธุ์	ส่วนสูง	น้ำหนัก	ชนิด
461	มานิวลา	1.1	34.0	มาร
462	จินะคอยล์	1.2	180.0	ไฟฟ้า
463	เบโรเบลต์	1.7	140.0	ธรรมชาติ
464	โดไซดอน	2.4	282.8	ดิน

	ชนิด	พลังโจมตี	พลังป้องกัน
461	น้ำแข็ง	120	65
462	เหล็ก	70	115
464	หิน	140	130
465	พืช	100	125

	สายพันธุ์	ส่วนสูง	น้ำหนัก	ชนิด 1	ชนิด 2	พลังโจมตี	พลังป้องกัน
461	มานิวลา	1.1	34.0	มาร	น้ำแข็ง	120.0	65.0
462	จินะคอยล์	1.2	180.0	ไฟฟ้า	เหล็ก	70.0	115.0
463	เบโรเบลต์	1.7	140.0	ธรรมชาติ	NaN	NaN	NaN
464	โดไซดอน	2.4	282.8	ดิน	หิน	140.0	130.0

จะเห็นว่า 465 ซึ่งไม่มีในตารางซ้ายถูกตัดทิ้งไป แต่ 463 ซึ่งไม่มีในตารางขวาก็ยังคงถูกเก็บไว้ ส่วน "ชนิด" ซึ่งเป็นชื่อคอลัมน์ที่ซ้ำกันก็ถูกเติม 1 และ 2 ต่อท้าย

อ้างอิง

<http://sinhrks.hatenablog.com/entry/2015/01/28/073327> (<http://sinhrks.hatenablog.com/entry/2015/01/28/073327>)
<http://d.hatena.ne.jp/Tommy1/20131209/1386545113> (<http://d.hatena.ne.jp/Tommy1/20131209/1386545113>)
<http://d.hatena.ne.jp/Tommy1/20131210/1386631784> (<http://d.hatena.ne.jp/Tommy1/20131210/1386631784>)
<http://d.hatena.ne.jp/Tommy1/20131211/1386719956> (<http://d.hatena.ne.jp/Tommy1/20131211/1386719956>)
<http://pythondatascience.plavox.info/pandas/pandasでデータフレームを結合> (<http://pythondatascience.plavox.info/pandas/pandasでデータフレームを結合>)
<http://d.hatena.ne.jp/civilclothes/20140529/1401357962> (<http://d.hatena.ne.jp/civilclothes/20140529/1401357962>)
<http://qiita.com/ynakayama/items/358a7e043194cecf28f9> (<http://qiita.com/ynakayama/items/358a7e043194cecf28f9>)



เรียงลำดับตาม เก้าที่สด

เพิ่มความคิดเห็น...

ดุษฎีของหน้านี้ (/dusathiti/yancham13)

หมวดหมู่ (/muat)

-- คอมพิวเตอร์

(/muat/%E0%B8%84%E0%B8%AD%E0%B8%A1%E0%B8%9E%E0%B8%B4%E0%B8%A7%E0%B9%80%E0%B8%95%E0%B8%AD%E0%B8%A3%E0%B9%8C

>> เขียนโปรแกรม

(/muat/%E0%B9%80%E0%B8%82%E0%B8%B5%E0%B8%A2%E0%B8%99%E0%B9%82%E0%B8%9B%E0%B8%A3%E0%B9%81%E0%B8%81%E0%B8%A3%E0%

```
>> python (/muat/python) >> pandas (/muat/pandas)
```

ไม่อนุญาตให้นำเนื้อหาของบริษัทไปลงที่อื่นโดยไม่ได้อุญาตโดยเด็ดขาด หากต้องการนำบางส่วนไปลงสามารถทำได้โดยไม่ต้องไม่ใช้การก๊อปปี้แปะแต่ให้เปลี่ยนคำพูดเป็นของตัวเอง หรือไม่ก็เขียนในลักษณะการยกข้อความอ้างอิง และไม่ว่ากรณีไหนก็ตาม ต้องให้เครดิตพร้อมใส่ลิงก์ของทุกบทความที่มีการใช้เนื้อหาเสมอ