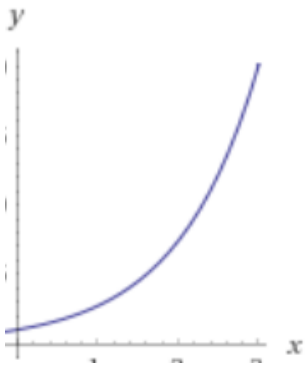
# 商店插件价格计算说明

服务器物品出售价格

物品出售价格由三部分组成：

1. 物品基础价格（取决于该物品的售出数）：如右图所示




```
current_base_price = item.base_price *
(item.base_price_increase_rate ** (item.sold_count /
item.sold_multiplier))
```

item.base_price：物品基础价，待填写参数

item.base_price_increase_rate：物品基础价乘数，一般为略微大于 1 的一个值

item.sold_count：物品售出数

item.sold_multiplier：物品售出系数

2. 市场均衡价（取决于该物品售出量占市场总物品的比重）

```
current_sold_base_price = current_base_price * (
        1 + item.sold_count / item.sold_multiplier / (total_sold + 1))
```

total_sold：物品含权重总售出数量

3. 含时恢复价格

```
a = item.half_time_recover / item.time_scale + 1
```

item.half_time_recover：物品从最低价到达最高价格恢复速度的时间

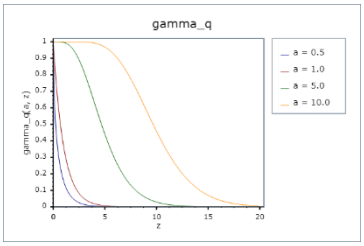
item.time_scale：时间因数，该项越高，恢复速度越慢

```
delta_time = current_time - item.last_sold_time
```

时间差，就是现在的时间和上一次售出的时间差

```
delta_time_normalized = delta_time / item.time_scale
```

除掉时间因数




```
depriced_percentage = sc.gammaincc(a,
delta_time_normalized)
```

不完全 gamma 函数（如右图），表示物品从上一个次出售后的涨到的最高价到目前的目标底价（第二段的结果）恢复的比例，返回一个 1~0 的值

```
before_decrease_price = item.last_sold_price * item.sold_price_multiplier
```

上一次出售后涨到的最高价

```
if (before_decrease_price < current_sold_base_price):
    return current_sold_base_price
current_price = (before_decrease_price - current_sold_base_price) *
depriced_percentage + current_sold_base_price
```

服务器物品回收价格

物品回收价格同样由三部分组成：最高价，市场均衡价和含时恢复函数

1. 最高价计算

```
reduced_max_price = math.exp(-item.max_price_reduce_rate *
item.bought_count/item.bought_multiplier) * (
            item.max_price - item.protected_max_price) +
item.protected_max_price
```

item.max_price_reduce_rate：最高价削减速率，一般是个极小的值

item.max_price：商品最高价

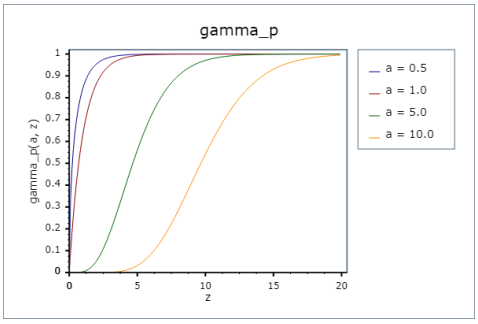
item.protected_max_price：商品最高价保护限价（一般为非负值）

2. 市场均衡价计算

```
    reduced_bought_max_price = (1 - item.bought_count /
    item.bought_multiplier / (
            1 + total_bought)) * reduced_max_price
```

3. 含时保护价

```
a = item.half_time_recover / item.time_scale + 1
last_price_decreased = item.last_price * item.bought_price_multiplier
pb = (last_price_decreased - item.lowest_price) / (reduced_bought_max_price
- item.lowest_price)
if (pb >= 1):
    pb = 1 - 1e-8
```




pb 代表的是商品的当前价位在最高价-最低价之间的位置

```
tb = sc.gammaincinv(a, pb)
```

tb 代表的是它在 gamma 函数中的时序，如图

```
delta_time = current_time -
item.last_bought_time
delta_time_normalized = delta_time /
item.time_scale
current_price = sc.gammainc(a, tb + delta_time_normalized) * (
            reduced_bought_max_price - item.lowest_price) +
item.lowest_price
```

再从时序还原得到物品的当前价格。