

Функции для работы с клиентом и картой

- [age](#) – Проверка возраста клиента
- [hasClientOptions](#) – Наличие опции у клиента
- [birthdayDate](#) – В день рождения
- [birthdayDateIn](#) – День рождения в интервале
- [specialDate](#) – В памятную дату
- [specialDateIn](#) – Памятная дата в интервале
- [noCardInCheck](#) – Отсутствие карты в чеке
- [isVerificated](#) – Признак верификации карты
- [hasVerificationTypes](#) – Наличие способов верификации

Системы лояльности торгового предприятия чаще всего направлены на определенную группу людей, объединенных некоторыми характеристиками. Скидки и бонусы в этом случае предоставляются для определенной категории покупателей. Функции для работы с клиентом и картой позволяют персонифицировать скидки, организуя акции направленного действия.

age – Проверка возраста клиента

`age()` -- функция возвращает количество полных лет, если указана дата рождения клиента, 0 – если дата не указана.

Пример

```
cf.age() >= 63
```

Скидка работает, если возраст клиента больше или равен 63 (пенсионер).

hasClientOptions – Наличие опции у клиента

`hasClientOptions([options])` – функция работает, если клиенту назначена указанная опция.

Параметры функции:

- `options` – опция клиента.

Опция клиента – это признак клиента, который задается при добавлении клиента в базу, например, блондинка, студент и т. д.



Применение функции возможно только при наличии в чеке карты клиента.

Пример

```
cf.hasClientOptions([u"блондинка"])
```

Скидка работает, если клиент имеет опцию "блондинка".

birthdayDate – В день рождения

`birthdayDate(firstDay,lastDay)` – функция работает, если день рождения клиента входит в заданный интервал.

Параметры функции:

- `firstDay` – количество дней до дня рождения;
- `lastDay` – количество дней после дня рождения.



Применение функции возможно только при наличии в чеке карты клиента.

Пример

```
cf.birthdayDate(2,3)
```

Скидка будет действовать, если чек открыт в день рождения клиента, либо 2 дня до и 3 дня после дня рождения.

birthdayDateIn – День рождения в интервале

`birthdayDateIn(beginDate, endDate)` – функция работает, если день рождения клиента входит в указанный диапазон дат.

Параметры функции:

- `beginDate` – начальная дата интервала;
- `endDate` – конечная дата интервала.



Применение функции возможно только при наличии в чеке карты клиента.

Пример

```
cf.birthdayDateIn("2014-1-1","2014-1-31")
```

Скидка действует, если день рождения клиента в январе. Время открытия чека не проверяется.

specialDate – В памятную дату

`specialDate(firstDay, lastDay, datename)` – функция работает, если чек открыт в памятную дату клиента или в заданный диапазон дней до/после нее. Специальная дата задается для каждого клиента индивидуально.

Параметры функции:

- `firstDay` – количество дней до даты;
- `lastDay` – количество дней после даты;
- `datename` – название специальной даты.



Применение функции возможно только при наличии в чеке карты клиента.

Пример

```
cf.specialDate(1,1,u"specialdate1")
```

Скидка для клиента действует, если чек открыт в "специальную дату 1" и 1 день до и после нее.

specialDateIn – Памятная дата в интервале

`specialDateIn(beginDate, endDate, datename)` – функция работает, если специальная дата клиента входит в указанный диапазон дат.

Параметры функции:

- `beginDate` – начальная дата интервала;
- `endDate` – конечная дата интервала;
- `datename` – название специальной даты.



Применение функции возможно только при наличии в чеке карты клиента.

Пример

```
cf.specialDateIn("2014-3-1", "2014-3-31", "specialdate1")
```

Скидка с таким условием работает если клиент имеет специальную дату №1 в марте.

noCardInCheck – Отсутствие карты в чеке

`noCardInCheck()` – функция работает, если в чеке нет ни одной карты.

Пример

```
cf.noCardInCheck()
```

Скидка будет действовать, если в чеке нет ни одной карты.

isVerified – Признак верификации карты

`isVerified()` – функция работает в том случае, если карта прошла верификацию.

Пример

```
cf.isVerified()
```

Скидка будет действовать, если карта, добавленная в чек, прошла верификацию.

hasVerificationTypes – Наличие способов верификации

`hasVerificationTypes(["type"])` – функция работает в том случае, если у группы карт, к которой принадлежит введенная карта, имеются все перечисленные в параметре [способы верификации](#).

Параметры функции:

- `type` – способ верификации.

Если параметр `type` не задан, то функция проверит наличие каких-либо способов верификации группы карт:

- при наличии способов – функция работает;
- при отсутствии способов – функция не работает.

При работе функции атрибут верификации не учитывается.

Пример

```
cf.hasVerificationTypes(["SMS", "AUTO"])
```

Функция будет действовать, если для группы, к которой принадлежит карта, заданы способы верификации SMS и AUTO.

Пример поведения функции

Задана акция оплаты бонусами с условиями:

1. Разрешено оплачивать бонусами 10% от суммы чека
2. Задано условие `cf.hasVerificationTypes(["SMS", "AUTO"])`

Пример 1

При формировании чека в него добавлена карта. В группе карт указан способ верификации "SMS".

Функция проверяет способы верификации. Так как в группе карт указан только 1 из способов верификации, то при переходе в подытог сумма бонусов для оплаты рассчитана не будет. При попытке оплатить бонусами на экран кассира будет выведено сообщение "Невозможно потратить баллы".

Пример 2

При формировании чека в него добавлена карта. В группе карт указан способ верификации "SMS" и "AUTO".

Функция проверяет способы верификации. Так как в группе карт указаны оба способа верификации, то при переходе в подытог рассчитывается сумма бонусов для оплаты. Далее оплата баллами производится по алгоритму [верификации карты при оплате](#).